**IN THE UNITED STATES DISTRICT COURT**
**FOR THE EASTERN DISTRICT OF TEXAS**
**MARSHALL DIVISION**

| | |
|---|---|
| UNITED SERVICES AUTOMOBILE ASSOCIATION,<br><br>*Plaintiff and Counterclaim-Defendant,*<br><br>v.<br><br>REGIONS BANK<br><br>*Defendant and Counterclaim-Plaintiff.* | Case No. 2:25-cv-00082-JRG<br><br>**JURY TRIAL DEMANDED** |

**DEFENDANT AND COUNTERCLAIM PLAINTIFF REGIONS BANK'S P.R. 3-3 INVALIDITY AND PATENT-INELIGIBILITY CONTENTIONS**

# TABLE OF CONTENTS


Page(s)

**I. Introduction ..... 1**

**II. Reservations ..... 2**

**III. IDENTIFICATION OF PRIOR ART PURSUANT TO P.R. 3-3(A) ..... 6**

**A.** Priority Dates of the Asserted Patents ..... 7

**B.** Prior Art Patents / Patent Applications ..... 10

**C.** Prior Art Publications ..... 11

**D.** Prior Art Products, Systems, and/or Services ..... 13

**IV. Claim charts PURSUANT TO P.R. 3-3(C) ..... 16**

**V. Disclosures of Invalidity ..... 16**

**A.** Disclosure of Invalidity due to Anticipation ..... 16

**B.** Disclosure of Invalidity due to Obviousness ..... 17

1. The Claimed Concepts Were Known ..... 18
2. Obviousness Combinations ..... 91
3. Reasons to Modify, Extend, or Combine the Prior Art References ..... 95
4. No Objective Indicia of Non-Obviousness ..... 229

**VI. Contentions under 35 U.S.C. § 112 ..... 229**

**A.** Lack of Written Description ..... 230

**B.** Enablement ..... 231

**C.** Indefiniteness ..... 233

**VII. Subject Matter Ineligibility ..... 234**

**A.** The ’310 Patent ..... 237

**B.** The ’095 Patent ..... 241

## I. INTRODUCTION

Pursuant to Rule 3-3 of the Rules of Practice for Patent Cases before the United States District Court for the Eastern District of Texas ("P.R."), the Court's Standing Order Regarding Subject Matter Eligibility Contentions ("Order Regarding Eligibility"), and the Court's May 30, 2025 Docket Control Order (ECF 32), Defendant and Counterclaim-Plaintiff Regions Bank ("Regions") hereby serves Invalidity Contentions and Patent-Ineligibility Contentions ("Invalidity Contentions") with respect to the asserted claims of 12,159,310 (the "'310 Patent") and 12,211,095 (the "'095 Patent") (together, the "Patents-in-Suit" or "Asserted Patents") identified by Plaintiff and Counterclaim-Defendant United Services Automobile Association ("USAA") in its Initial Disclosure of Asserted Claims, Infringement Contentions, And Accompanying Document Production ("Infringement Contentions" or "P.R. 3-1 and 3-2 disclosures").

USAA has asserted the following claims:

- '310 Patent: Claims 1-16
- '095 Patent: Claims 1-30

With respect to each asserted claim and based on its investigation to date, Regions hereby: (a) identifies each prior art reference that anticipates each asserted claims or renders it obvious according to P-R 3-3(a); (b) specifies whether each such prior art reference anticipates each asserted claim or renders it obvious, and, if it renders it obvious, identifies any combinations of prior art showing obviousness and explains the motivation to combine the prior art that renders the asserted claim obvious; (c) submits a chart identifying where specifically in each prior art reference each element of each asserted claim is found, including, for each element that is governed by 35 U.S.C. § 112(6), the identity of the structure(s), act(s), or material(s) in

each prior art reference that performs the claimed function; (d) identifies the grounds of invalidity based indefiniteness under 35 U.S.C. § 112(2) or enablement or written description under 35 U.S.C. § 112(1) of any of the asserted claims; and (e) identifies the basis for its defense that the asserted patents are invalid under 35 U.S.C. § 101.

In addition, pursuant to P.R. 3-4(a), Regions has produced documents and source code in its possession showing the operation of the accused functionality. Pursuant to P.R. 3-4(b), Regions has produced each item of prior art identified pursuant to P.R. 3-3(a), which does not appear in the file history of the patents-in-suit.

## II. RESERVATIONS

Consistent with P.R. 3-6, Regions reserves the right to further amend and/or supplement these Invalidity Contentions. The information and documents that Regions produces are provisional and subject to revision. Regions expressly reserves the right to amend these disclosures should USAA provide any information that it failed to provide in its Infringement Contentions, or should USAA amend its P.R. 3-1 or 3-2 disclosures in any way. Further, because discovery is only beginning, Regions reserves the right to revise, amend, and/or supplement the information provided herein, including identifying and relying on additional references, should Regions further search and analysis yield additional information or references, consistent with the Patent Local Rules and the Federal Rules of Civil Procedure, including any further information or references revealed by any third party subpoenas that have not yet been served. Moreover, Regions reserves the right to revise its ultimate contentions concerning the invalidity of the asserted claims, which may change depending upon the Court's construction of the asserted claims, any evidence or findings as to the priority or invention date of the asserted

claims, and/or positions that USAA or its expert witness(es) may take concerning claim construction, infringement, and/or invalidity issues.

Prior art not included in this disclosure, whether known or unknown to Regions, may become relevant. In particular, Regions is currently unaware of the extent, if any, to which USAA will contend that limitations of the asserted claims are not disclosed in the prior art identified by Regions, or will contend that any of the identified references do not qualify as prior art. To the extent that such issues arise, Regions reserves the right to identify additional teachings in the same references or in other references that anticipate or would have made the addition of the allegedly missing limitation to the claimed device obvious, consistent with the Patent Local Rules. In providing these contentions, Regions has relied on the contents of USAA's P.R. 3-1 and 3-2 disclosures served on May 14, 2025.

Regions' claim charts in Exhibits A1 through C2 cite to particular teachings and disclosures of the prior art as applied to features of the asserted claims. However, persons having ordinary skill in the art may view an item of prior art generally in the context of other publications, literature, products, and understanding that was available in the art. Accordingly, the cited portions are only examples, and Regions reserves the right to rely on uncited portions of the prior art references and on other publications and expert testimony as aids in understanding and interpreting the cited portions, as providing context thereto, and as additional evidence that a claim limitation is known or disclosed. Citations to figures are inclusive of all discussion of those figures. Citations to descriptions of elements present in a figure are inclusive of all figures containing such elements. Regions further reserves the right to rely on uncited portions of the prior art references, other publications, documents explicitly or implicitly incorporated by reference, and testimony, to establish bases for combinations of certain cited references that

render the asserted claims obvious. Further, for any combination, Regions reserves the right to rely additionally on information generally known to those skilled in the art and/or common sense.

The references discussed herein and in the claim charts in Exhibits A1 through C2, or elsewhere identified, may disclose the elements of the asserted claims explicitly and/or inherently, and/or they may be relied upon to show the state of the art in the relevant timeframe. The suggested obviousness combinations are provided in the alternative to Regions' anticipation contentions and are not to be construed to suggest that any reference included in the combinations is not itself anticipatory.

Each reference cited herein and in the claim charts in Exhibits A1 through C2, or elsewhere identified, and relied upon by Regions as a patent or printed publication under 35 U.S.C. §§ 102(a) or (b) should also be understood to be relied upon by Regions to establish that the disclosure therein was "known or used by others" under § 102(a) and/or "in public use or on sale" under § 102(b), and otherwise known for purposes of evaluating patent-ineligibility under § 101. Regions reserves the right to establish that the disclosure in any reference was known or used by others and/or in public use or on sale, irrespective and independent of the reference's disclosure as a patent or a printed publication.

Regions further reserves the right to assert that the asserted claims are invalid under 35 U.S.C. § 102(f) in the event that Regions obtains evidence that the named inventors of the Asserted Patents did not invent (either alone or in conjunction with others) the subject matter recited in the asserted claims. Should Regions obtain such evidence, it will provide the name of the person(s) from whom, and the circumstances under which, the invention or any part of it was derived. Regions further intends to rely on inventor admissions concerning the scope of the

asserted claims or of the prior art relevant to the asserted claims found in, *inter alia*: the patent prosecution history for the Asserted Patents and related patents and/or patent applications; any deposition testimony of the named inventors of the Asserted Patents; and the papers filed and any evidence submitted by USAA in conjunction with this litigation or any related actions. To the extent any information is identified under § 102(f), Regions reserves the right to contend that the patent is invalid for failure to name the correct inventor, and/or to contend that Regions lacks standing to bring this litigation with respect to such patent.

Furthermore, nothing stated herein or in Exhibits A1 through C2 shall be treated as an admission or suggestion that Regions agrees with USAA regarding either the scope of any asserted claim or the claim constructions USAA advances in its P.R. 3-1 and 3-2 disclosures or anywhere else. To the extent that Regions' Invalidity Contentions reflect constructions of claim limitations consistent with or suggested by USAA's P.R. 3-1 and 3-2 disclosures, no inference is intended nor should any be drawn that Regions agrees with USAA's claim constructions. Nor shall anything in these Invalidity Contentions be treated as an admission that Regions' accused technology meets any limitation of any asserted claim. Regions denies that it infringes any claim of the Asserted Patents. To the extent that any prior art reference identified by Regions contains a claim element that is the same as or similar to an element in an accused product, based on a claim construction inferred from USAA's P.R. 3-1 and 3-2 disclosures, inclusion of that reference in Regions' Invalidity Contentions shall not be deemed a waiver by Regions of any claim construction or noninfringement position. Regions expressly reserves the right to contest any claim construction asserted by USAA and expressly reserves all noninfringement arguments.

Depending on the Court's construction of the asserted claims of the Asserted Claims, and/or positions that USAA or its expert witness(es) may take concerning claim interpretation,

infringement, and/or invalidity issues, different ones of the charted prior art references in Exhibits A1 through C2, or otherwise identified herein, may be of greater or lesser relevance and different combinations of these references may be implicated. Given this uncertainty, the charts may reflect alternative applications of the prior art against the asserted claims. Moreover, given this uncertainty, Regions also reserves the right to supplement with other references or additional citations to existing references. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Regions also reserves all rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description, and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any additional arguments available under 35 U.S.C. § 112.

Regions reserves the right to establish that products identified in Section III.D embody one or more claims of the Asserted Patents and such products were publicly used, offered for sale, or on sale in the United States more than a year before the priority date(s) of the Asserted Patents, as discussed below in Section III.A.

## III. IDENTIFICATION OF PRIOR ART PURSUANT TO P.R. 3-3(A)

In addition to the disclosures provided below, Regions attaches separate charts detailing its invalidity contentions on a claim-by-claim basis. *See* Exhibits A through B24. Moreover, Regions hereby discloses and reserves the right to rely upon all references incorporated by reference into the references disclosed below and in the accompanying charts.[1] For the avoidance of doubt (where not separately identified), identification of a granted patent by patent

[1] To the extent a cited reference is found not to predate any Asserted Patent for which it is cited, it remains relevant, for example, to show the technological background and/or objective indicia of obviousness and/or obviousness based on simultaneous/contemporaneous invention by others.

number also identifies the corresponding publication numbers and provisional application numbers, and vice versa.

### A. Priority Dates of the Asserted Patents

USAA alleges that "the Asserted Claims of the '310 Patent and '095 Patent are entitled to priority dates of at least as early as, and possibly earlier than August 21, 2009 (Filing Date of Appl. No. 12/545,127)." May 14, 2025 Infringement Contentions at 19. USAA has not established that the '310 and '095 Patents are entitled to claim the benefit of that application. Absent such evidence, the '310 and '095 patents are not entitled to claim priority to August 21, 2009. Indeed, as explained in Section VI.A and VI.B, the Asserted Claims of the '310 and '095 Patents lack written description and are not enabled by the disclosure in USAA's patent specification, and are therefore not entitled to a priority date any earlier than their respective filing dates.

USAA further alleges that "the inventions disclosed in the '310 and '095 Patents were conceived by at least July 2008 in connection with research and development work by USAA's Applied Research and IT Technology teams related to USAA's Deposit@Home and Deposit@Mobile Products, and these inventions were diligently reduced to practice by at least the filing dates of the 2006 and 2009 priority applications discussed above." May 14, 2025 Infringement Contentions at 19-20. For the reasons outlined in Regions June 20, 2025 Letter from Andrew Danford to Lisa Glasser, USAA's disclosures pursuant to P.R. 3-2 are deficient for numerous reasons. For example, USAA did not provide the metadata required to evaluate the date various documents in USAA's P.R. 3-2 disclosure. Moreover, none of the documents in USAA's P.R. 3-2 disclosure demonstrate conception or reduction to practice numerous features

alone or in combination with the other features in the ’310 and ’095 patent claims or in the claimed ordering of steps, including:

- A downloaded application (e.g., ’310 patent, claim 1[a], 1[c], 1[k], 1[m], 2[a], 2[c], 2[k], 5[a], 5[c], 6[a], 6[c], 8[a], 8[c], 8[k], 9[a], 9[c], 10[a], 10[c], 10[j], 11[a], 11[c], 14[a], 14[c], 14[k], 15[a], and 15[c]; ’095 patent, claims 1[c], 1[e], 2[a], 9[a], 9[c], 18[c], 18[e], 19[a], 26[c], 26[e], and 27[a]);

- Software abstraction layers (e.g., ’310 patent, claim 1[a], 1[c], 2[a], 2[c], 5[c], 6[c], 8[c], 9[c], 10[a], 10[c], 11[a], 11[c], 14[a], 14[c], 15[a], and 15[c]; ’095 patent, claims 1[b], 1[c], 1[e], 18[b], 18[c], 18[e], 26[b], 26[c], and 26[e]);

- Wirelessly submitting checks for deposit (’310 patent, claims 1[h], 2[h], 5[e], 6[e], 8[e], 9[e], 10[g], 11[g], 14[g], and 15[g]; ’095 patent, claims 1[j], 9[h], 18[i], and 26[i]);

- Monitoring live video (’310 patent, claims 1[e], 2[e], 5[a], 6[a], 8[a], 9[a], 10[d], 11[d], 14[d], and 15[d]; ’095 patent, claims 1[g], 9[e], 18[f], and 26[f]);
- Monitoring criteria such as brightness or MICR detection (’310 patent, claims 2[m], 3, 4, 6[k], 7, 11[j], 12, 13, 15[k], and 16; ’095 patent, claims 3, 4, 5, 12, 13, 20, 21, 22, 28, and 29);
- Confirming check deposit can go forward (including before submission) (’310 patent, claims 1[i], 1[k], 1[l], 2[i], 2[k], 2[l], 5[i], 5[j], 6[i], 6[j], 8[i], 8[j], 9[i], 9[j], 10[h], 10[i], 11[h], 11[i], 14[h], 14[j], 15[h], and 15[j]; ’095 patent, claims 1[k], 1[n], 1[o], 6, 9[j], 9[k], 14, 18[j], 18[k], 23, 26[j], and 26[l]);

- Performing OCR (’310 patent, claims 1[i], 1[k], 2[i], 2[k], 5[i], 6[i], 8[i], 9[i], 10[h], 11[h], 14[h], and 15[h]; ’095 patent, claims 1[k], 1[n], 9[i], 9[j], 18[j], 26[j], and 26[k]);
- Recognizing a MICR line (’310 patent, claims 1[i], 2[i], 5[i], 6[i], 8[i], 9[i], 10[h], 11[h], 14[h], and 15[h]; ’095 patent, claims 1[k], 9[j], 18[j], and 26[j]);
- Validating an amount payable (’310 patent, claims 1[i], 2[i], 5[i], 6[i], 8[i], 9[i], 10[h], 11[h], 14[h], and 15[h]; ’095 patent, claims 1[k], 9[j], 18[j], and 26[j]);
- Indication of a result of the processing (’310 patent, claims 1[k], 2[k], and 9[k]; ’095 patent, claim 1[n] and 15);
- Updating an account balance (’310 patent, claims 1[l], 2[l], 5[j], 6[j], 8[j], 9[j], 10[i], 11[i], 14[j], and 15[j]; ’095 patent, claims 1[o], 9[k], 18[k], and 26[l]);
- Overlaying an alignment guide on live video (’310 patent, claims 1[d], 2[d], 5[d] 6[d], 8[d], and 9[d]; ’095 patent, claims 1[f] and 9[d]);
- Specific combinations of monitoring criteria (’310 patent, claims 2[m], 3, 4, 6[k], 7, 11[j], 12, 13, 15[k], and 16; ’095 patent, claims 3, 4, 12, 20, 21, 28, and 29);

Indeed, the sparsity of the ’310 and ’095 patents’ disclosure (*see* Sections VI.A, VI.B) confirms that USAA neither conceived of nor reduced to practice the purported invention(s) of the patents before their filing dates. Accordingly, USAA the ’310 and ’095 Patents are not entitled to a priority date earlier than their respective filing dates.

Moreover, the ’095 patent as published does not present any claim of priority to any earlier application—either on its cover or in the specification. The ’095 patent is therefore not entitled to a priority date any earlier than its filing date of March 1, 2024. USAA’s Infringement Contentions allege that functionality in Regions’ mobile application infringe the claims of the

'095 patent, but such functionality predates March 1, 2024. While Regions denies that its mobile application infringes any claim of the '095 patent, by USAA's own admission, any allegation of infringement means that the '095 patent is anticipated by Regions' mobile application, which was on sale and publicly used prior to March 1, 2024.[2]

**B. Prior Art Patents / Patent Applications[3]**

The following prior art patents and patent applications, including those references listed in Exhibits A1 through B13, anticipate and/or render obvious the Asserted Claims of the Asserted Patents. Any references not expressly included in Exhibits A1 through B13 may be relied upon to demonstrate the state of the art and/or the knowledge of a person of ordinary skill in the art as it relates to the Asserted Patents in the relevant timeframe.

| Exhibit(s) | Patent No. / Application No. | Filing Date | Issue or Publication Date | Short Name | Production No. |
|---|---|---|---|---|---|
| A1, B1 | US 8,768,836 | August 7, 2007 | July 1, 2014 | Acharya | REGIONS_USAA-00002085 |
| A2, B2 | US 2005/0002666 | March 26, 2004 | January 6, 2005 | Asai | REGIONS_USAA-00002574 |
| A3, B3 | US 2006/0249567 | February 9, 2006 | November 9, 2006 | Byrne | REGIONS_USAA-00002361 |
| A4, B4 | US 2007/0194102 | February 18, 2006 | August 23, 2007 | Cohen | REGIONS_USAA-00002347 |
| A5, B5 | US 2010/0205090 | February 10, 2009 | August 12, 2010 | Fellerman | REGIONS_USAA-00002239 |

[2] USAA filed for a certificate of correction of the '095 patent on May 13, 2025, which has not been issued. To the extent that such certificate issues, it would only be effective for causes of action arising after the certificate's issue date, and thus cannot be considered in this case. *H-W Tech., L.C. v. Overstock.com, Inc.*, 758 F.3d 1329, 1334 (Fed. Cir. 2014).

[3] To the extent USAA contends that a reference is not prior art under § 102(e), Regions reserves the right to respond by showing the support for a claim in the earlier filed patent application analysis. *See Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375 (Fed. Cir. 2015)



USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 12 of 1772

| Exhibit(s) | Patent No. / Application No. | Filing Date | Issue or Publication Date | Short Name | Production No. |
|---|---|---|---|---|---|
| A6, B6 | WO 2005/043857 | October 30, 2003 | May 12, 2005 | Garcia | REGIONS_USAA-00001915 |
| A7, B7 | US 2010/0082444 | September 30, 2008 | April 1, 2010 | Lin | REGIONS_USAA-00002254 |
| A8, B8 | CN 1897644A | July 15, 2005 | May 12, 2010 | Luo | REGIONS_USAA-00002820 |
| A9, B9 | US 7,978,900 | December 30, 2008 | July 12, 2011 | Nepomniachtchi | REGIONS_USAA-00002131 |
| A10, B10 | US 2005/0131820 | December 11, 2003 | June 16, 2005 | Rodriguez | REGIONS_USAA-00002527 |
| A11, B11 | US 8,131,636 | May 5, 2005 | March 6, 2012 | Viera | REGIONS_USAA-00002100 |
| A12, B12 | US 2007/0262148 | May 3, 2007 | November 15, 2007 | Yoon | REGIONS_USAA-00002339 |
| A13, B13 | CN 1967565A | November 22, 2006 | October 22, 2008 | Zhu | REGIONS_USAA-00002806 |

### C. Prior Art Publications

The following non-patent publications, including those references listed in Exhibits A14 through B20, are prior art to the asserted claims under at least pre-AIA 35 U.S.C. §§ 102(a), (b), (e), and/or (g) and/or pre-AIA 35 U.S.C. § 103 and anticipate and/or render obvious the Asserted Claims of the Asserted Patents. Any references not expressly included in Exhibits A14 through B18 may be relied upon to demonstrate the state of the art and/or the knowledge of a person of ordinary skill in the art as it relates to the Asserted Patents in the relevant timeframe.



USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 13 of 1772

| Exhibit(s) | Title | Date of Earliest Publication[4] | Author(s) | Short Name | Bates |
|---|---|---|---|---|---|
| A14, B14 | Android Website: Class Index, android.hardware.CameraDevice;<br><br>Android – An Open Handset Alliance Project | December 25, 2007 | Android - An Open Handset Alliance Project | Android Website, Andorid Website (android.hardware.CameraDevice), Android Website (Open Handset) | REGIONS_USAA-00004262;<br><br>REGIONS_USAA-00005730 |
| A15, B15 | Android Developers Blog-An Open Handset Alliance Project: Android Market: Now Available for Users, Android Developers Website: Camera | October 22, 2008, | Android Developers Website | Android Developers Website, Android Developers Website (Camera), Android Developers Website (Android Market) | REGIONS_USAA-00004262;<br><br>REGIONS_USAA-00005727 |
| | Mobile Design and Development | August 2009 | Brian Fling | Fling | REGIONS_USAA-00003828 |
| A16, B16 | Pro Java ME MMAPI | 2006 | Vikram Goyal | Goyal | REGIONS_USAA-00002852 |
| A17, B17 | Professional Android Application Development | November 2008 | Reto Meier | Meier | REGIONS_USAA-00003147 |
| A18, B18 | Using a Camera with Windows Mobile 5 | July 21, 2006 | Amit Ranjan | Ranjan | REGIONS_USAA-00003130 |
| | Mobile Development Handbook | 2007 | Andy Wigley, Daniel Moth, and Peter Foot | Wigley | REGIONS_USAA-00003790 |

[4] Regions reserves the right to rely on additional evidence, including but not limited to additional documents, versions, websites (including pages from the Internet Archive), and witness testimony, to establish the publication date of each prior art publication, including additional publication dates after the identified date of earliest publication.



USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 14 of 1772

### D. Prior Art Products, Systems, and/or Services

The following products, systems, and services, including those references listed in Exhibits A19 through B24, are prior art to the Asserted Patents under at least 35 U.S.C. §§ 102(a), (b) and/or (g) and anticipate and/or render obvious the Asserted Claims of the Asserted Patents. Regions contends that the Asserted Patents are invalid in view of public knowledge and uses and/or offers for sale or sales of the below products, systems and services, at least when applying USAA's apparent interpretation of the claims in its P.R. 3-1 and 3-2 disclosures. Regions' investigation of these sources of prior art is ongoing. Regions identifies, to the extent now known, when each item became publicly known or was used, offered for sale, or sold. Regions contends that the below are stated on information and belief, and are supported by the information and documents that will be produced by Regions and/or third parties. As discovery has just begun, Regions continues to investigate these events and reserves the right to add additional prior art products, systems, and services and/or to rely on additional documents or portions of documents describing the products, systems, and services below, including any documents produced in response to a subpoena, in order to demonstrate the functionality of the products, systems, and services identified below as well as the date by which such products, systems, and services were publicly known or used, offered for sale, or sold.

- Products, components, systems, services, and methods invented, designed, developed, reduced to practice, and/or in public use or on sale related to the CAM Mobile Document Processing System ("CAM"), as exemplified in Exhibits A19, B19. As part of these Invalidity Contentions, Regions has produced documents relating to the CAM Mobile Document Processing System. Regions believes that the CAM Mobile Document Processing System was conceived and/or reduced to practice by Dr. Tapan S. Parikh and his colleagues at least before October 23,

2005, without being abandoned, suppressed, or concealed, and it was known or used by others in this country or in public use or on sale on by the dates identified in the claim chart, which are no later than October 23, 2005.

- Products, components, systems, services, and methods invented, designed, developed, reduced to practice, and/or in public use or on sale related to the ImageNet Mobile Deposit system ("ImageNet" or "Mitek"), as exemplified in Exhibits A20, B20. As part of these Invalidity Contentions, Regions has produced documents relating to the ImageNet Mobile Deposit system. Regions believes that the ImageNet Mobile Deposit system was conceived and/or reduced to practice by engineers at Mitek Inc. at least before January 2008, without being abandoned, suppressed, or concealed, and it was known or used by others in this country or in public use or on sale on by the dates identified in the claim chart, which are no later than January 2008.

- Products, components, systems, services, and methods invented, designed, developed, reduced to practice, and/or in public use or on sale related to the Rokr E6 Mobile Phone device ("MotoRokr"), as exemplified in Exhibits A21, B21. As part of these Invalidity Contentions, Regions has produced documents relating to the Rokr E6 Mobile Phone device. Regions believes that the Rokr E6 Mobile Phone device was conceived and/or reduced to practice by engineers at Motorola Mobility LLC at least before November 14, 2006, without being abandoned, suppressed, or concealed, and it was known or used by others in this country or in public use or on sale on by the dates identified in the claim chart, which are no later than December 4, 2006.

- Products, components, systems, services, and methods invented, designed, developed, reduced to practice, and/or in public use or on sale related to the Occipital RedLaser Mobile Phone App ("RedLaser"), as exemplified in Exhibits A22, B22. As part of these Invalidity Contentions, Regions has produced documents relating to the Occipital RedLaser Mobile Phone App. Regions believes that the Occipital RedLaser Mobile Phone App was conceived and/or reduced to practice by engineers at Occipital Inc. at least before June 2009, without being abandoned, suppressed, or concealed, and it was known or used by others in this country or in public use or on sale on by the dates identified in the claim chart, which are no later than June 2009.

- Products, components, systems, services, and methods invented, designed, developed, reduced to practice, and/or in public use or on sale related to the West Virginia United Federal Credit Union Mobile Phone Application ("WVUFCU App"), as exemplified in Exhibits A23, B23. As part of these Invalidity Contentions, Regions has produced documents relating to the West Virginia United Federal Credit Union Mobile Phone Application. Regions believes that the West Virginia United Federal Credit Union Mobile Phone Application was conceived and/or reduced to practice by engineers at West Virginia United Federal Credit Union at least before July 4, 2009, without being abandoned, suppressed, or concealed, and it was known or used by others in this country or in public use or on sale on by the dates identified in the claim chart, which are no later than July 4, 2009.

- Products, components, systems, services, and methods invented, designed, developed, reduced to practice, and/or in public use or on sale related to the ScanZoom Mobile Phone Application ("ScanZoom"), as exemplified in Exhibits A24, B24. As part of these Invalidity Contentions, Regions has produced documents relating to the ScanZoom Mobile Phone Application. Regions believes that the ScanZoom Mobile Phone Application was conceived and/or reduced to practice by engineers at Scanbuy, Inc. at least before October 11, 2004, without being abandoned, suppressed, or concealed, and it was known or used by others in this country or in public use or on sale on by the dates identified in the claim chart, which are no later than October 11, 2004.

## IV. CLAIM CHARTS PURSUANT TO P.R. 3-3(C)

Individual claim charts that identify where each element of each asserted claim can be found in each item of prior art are attached hereto. A listing of these claim charts is provided below:

- Exhibits A1 through A24: Claim charts for the '310 patent
- Exhibits B1 through B24: Claim charts for the '095 patent

## V. DISCLOSURES OF INVALIDITY

### A. Disclosure of Invalidity due to Anticipation

Regions reserves the right to present anticipation grounds based at least in part on the Court's construction of the asserted claims, any evidence or findings as to the priority or invention date of the asserted claims, any evidence that results from discovery, and/or positions that USAA or its expert witness(es) may take concerning claim construction, infringement, and/or invalidity issues.

### B. Disclosure of Invalidity due to Obviousness

Subject to the reservation of rights above and based on Regions' present understanding of the Asserted Claims of the Asserted Patents, and the apparent constructions USAA is asserting based on its P.R. 3-1 and 3-2 disclosures, the prior art references identified above in Section III, and charted in Exhibits A1 through B24, each anticipate or render obvious the asserted claims, either alone or in combination. To the extent a finder of fact finds that a limitation of a given claim was not disclosed by one of the references identified above, those claims are nevertheless unpatentable as obvious because the asserted claims contain nothing that goes beyond ordinary innovation. To the extent not anticipated, no asserted claim goes beyond combining known elements to achieve predictable results or does more than choose between clear alternatives known to those of skill in the art, particularly in view of the state of the art as reflected in the relevant references listed above in Section III.

Moreover, to the extent the foregoing references are found not to anticipate the asserted claims, the foregoing references render the asserted claims obvious either alone or in combination with one or more of the other references herein. As explained herein and/or in the accompanying charts, it would have been obvious to a person of skill in the art at the time of the alleged invention(s) of the Asserted Claims of the Asserted Patents to combine the various references cited herein so as to practice the Asserted Claims of the Asserted Patents. In addition to the specific illustrative combinations of prior art and the specific illustrative combinations of groups of prior art disclosed, Regions reserves the right to rely on any other combination of any prior art references disclosed herein or in any amendment to Regions' Invalidity Contentions. Regions further reserves the right to rely upon combinations disclosed within the prosecution history of the references cited herein. These obviousness combinations reflect Regions' present

understanding of the potential scope of the claims that USAA appears to be advocating and should not be construed as Regions' acquiescence to USAA's interpretation of the patent claims.

Prior art references rendering the Asserted Claims of the Asserted Patents obvious, alone or in combination with other references, are discussed below and included in Exhibits A1 through B24. Further reasons to combine the references identified in Exhibits A1 through B24 include the nature of the problem being solved, the express, implied and inherent teachings of the prior art, the knowledge of persons of ordinary skill in the art, that such combinations would have yielded predictable results, and that such combinations would have represented known alternatives to a person of ordinary skill in the art.

To be clear, the charts below do not necessarily indicate all instances where claim elements are discussed based on USAA's apparent interpretation of the claims as evidenced by its P.R. 3-1 and 3-2 disclosures. However, the omission or addition of indications based on these apparent interpretations is not to be construed as an admission as to the proper construction or scope of the claims. Regions reserves the right to challenge USAA's implicit or explicit construction of all claim terms.

### 1. The Claimed Concepts Were Known

The Asserted Claims of the Asserted Patents are purportedly directed to various aspects of mobile check deposit. As explained below, each of the concepts claimed in the Asserted Patents were well known before the patents' alleged priority date(s), especially in view of USAA's apparent construction of the claim terms as evidenced by its P.R. 3-1 and 3-2 disclosures. In addition, Exhibits A1 through B24 identify disclosures in the prior art for each of the elements of the Asserted Claims of the Asserted Patents, which also demonstrate that these concepts were well known.

#### a. Depositing a Check Using an Image ('310, '095 Patent)

Before the Asserted Patents, it was well known to deposit a check using an image taken of the check. For example:

- Garcia at 4:11-18 ("The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations."); 1:6-13 ("The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection."); 5:1-6:10; 9:10-10:16.

- Acharya at Abstract ("A system and method for initiating a deposit transaction, where the depositor is a banking customer located at a remote location, and where the item to be deposited is a financial instrument, such as a paper check from a third party, payable to the depositor. The enabling system features a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. An image and/or other data of the financial instrument are transmitted from the RCT to the Bank of First Deposit (BOFD) where the data may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction. The deposit transaction can be accomplished without physical transfer of the financial instrument to or through the bank system. The system and method provide convenience and improved transaction processing speed compared to other deposit transactions of financial instruments.").

- Lin at [0160] ("The check information also may be acquired through the camera. For example, the graphical element 236 may be selected to enter the check information using techniques similar to those described with respect to FIGS. 13 and 14. In certain embodiments, the check information may be acquired using the scanner 48 (FIG. 2). For example, the graphical element 336 may be selected to scan a bar code on the check using the scan device 48 (FIG. 2). In other embodiments, the payment information may be acquired by scanning an NFC enabled check. In these embodiments, a graphical element may be included within the summary bar 152 to enter the check information using near field communication. The check information may be acquired by near field communication using techniques similar to those described above with respect to FIGS. 11-12."); [0162] ("In certain embodiments, the approval step may include transmitting the check information to a verification service for approval. For example, the merchant may hire a third party check verification service to verify checks received from customers. In these embodiments, selection of the graphical element 228 may cause the device to send the check information to the external server 96 (FIG. 4), which may in turn send the check information to the check verification service. The check verification service may approve the check for

payment and transmit the approval to the electronic device 10 through the external server. Upon receipt, the payment authorization may be displayed within the window 242. The check information also may be transmitted to the customer's bank to verify that the customer has sufficient funds to process the transaction. Further, in other embodiments, the approval process may simply include verifying that all the check information has been entered, and no additional verification may be performed. After the transaction has been approved, the merchant may select the POP graphical element 224 to conduct another transaction.").

- Rodriguez at Abstract ("Methods, systems, and media for processing and clearing a check are disclosed. Embodiments include hardware and/or software for generating an e-check in response to receipt of a paper check, by which a merchant performs the manual data entry for the merchant, the merchant's bank, the Federal Reserve Bank, and the purchaser's bank clearing procedures. The paper check is electronically scanned at the point of sale (POS) terminal to create a check object (the 'e-check') that contains the binary data of the check information, a scanned image of the paper check, and identification data from the purchaser. Then, the check object is submitted to the check writer's, or purchaser's bank, for automated clearing without having to process the physical document any further.").

- Fellerman at [0020] ("In yet another embodiment of the present invention, an electronic deferred check payment issuance system is contemplated. The system may include a remote merchant terminal, and a check image scanner connected to the remote merchant terminal. The check image scanner may digitize a plurality of checks that each has an account identifier and a monetary value associated therewith. Furthermore, each of the checks may be tendered as partial payment for a single transaction between a merchant and a purchaser. The system may also include a central payment processor that is in communication with the remote merchant terminal. Additionally, the central payment processor may be in communication with a check clearing network to which a merchant account and a purchaser account designated by the account identifier on the check is also connected. The funds quantified by the monetary values of the checks may be deductible from the purchaser account and depositable to the merchant account by the central payment processor through the check clearing network. The deductions may proceed according to a set of schedules.").

- Viera at Abstract ("Embodiments of the present invention include a computer-implemented method of performing a financial transaction on a network comprising storing a plurality of risk factors and associated severities in a web-enabled remote policy database, the severities defining actions taken when a specific risk factor is detected, storing a plurality of hold override policies that associate a deposit type with a hold type in the policy database, receiving deposit data including an image of a deposit check, applying the plurality of risk factors to the deposit data, and applying the plurality of hold override policies to the deposit data.").

- Byrne at Abstract ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); *see also* FIGs. 1-7 (describing mobile deposit workflow), [0003], [0005]-[0011] (describing prior art relating to depositing a check using an image); [0033]; [0041]; [0172]-[0192] (describing exemplary client flow).

- Nepomniachtchi at 4:65–5:7 (The present invention is directed towards automated document processing and systems and methods for document image processing using mobile devices. Generally, some embodiments of the invention capture an original color image of a document using a mobile device and then converted the color image to a bi-tonal image. More specifically, some embodiments accept a color image of a check taken by a mobile device and convert it into a bi-tonal image of the check that is substantially equivalent in its resolution, size, and quality to check images produced by "standard" scanners); 5:60-6:7 (FIG. 1 is a diagram illustrating an example check 100 that might be imaged with the systems and methods described herein. The mobile image capture and processing systems and methods may be used with a variety of documents, including financial documents such as personal checks, business checks, cashier's checks, certified checks, warrants or other documents. By using an image of the check 100, the check clearing process is performed more efficiently. As would be appreciated by those of skill in the art, checks are not the only type of documents that may be used with these systems. For example, other documents, such as deposit slips, might also be processed using the systems and methods described herein. FIG. 2 is a diagram illustrating an example payment coupon 200 that might be imaged using the systems and methods described herein.); *see also id.* at 9:22–29; 23:11–12; 23:15–16.

- ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone").

- WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

**b. Depositing a Check Using an Image Captured By a Mobile Device ('310, '095 Patent)**

Before the Asserted Patents, it was well known to deposit a check using an image taken of the check by a mobile device, such as a personal digital assistant ("PDA") or a mobile phone. For example:

- Garcia at 4:11-18 ("The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations."); 1:6-13 ("The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection."); 5:1-6:10; 9:10-10:16.

- Acharya at Abstract ("A system and method for initiating a deposit transaction, where the depositor is a banking customer located at a remote location, and where the item to be deposited is a financial instrument, such as a paper check from a third party, payable to the depositor. The enabling system features a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. An image and/or other data of the financial instrument are transmitted from the RCT to the Bank of First Deposit (BOFD) where the data may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction. The deposit transaction can be accomplished without physical transfer of the financial instrument to or through the bank system. The system and method provide convenience and improved transaction processing speed compared to other deposit transactions of financial instruments."); 4:18 ("The RCT 100 may be a telephone, digital camera, fax machine, personal computer, ATM, cell phone, PDA or any other computer, apparatus, wireless handheld device such as a Blackberry® or Palm Treo® or system capable of collecting data and communicating with BOFD system 110.

For instance, the RCT 100 may comprise a workstation or other device running the Microsoft Windows™ NT™, Windows™ 2000, Windows Vista™, Windows XP™, Windows Mobile™, Unix, Linux, Xenix, IBM AIX, Hewlett-Packard UX, Novell Netware™, Sun Microsystems Solaris™, OS/2™, BeOS™, Mach, Apache, OpenStep™, Mac OS X™, GAME BOY™, PXP™ or any other operating system or platform now known or later developed. RCT 100 may also comprise two or more such devices.").

- Lin at [0160] ("The check information also may be acquired through the camera. For example, the graphical element 236 may be selected to enter the check information using techniques similar to those described with respect to FIGS. 13 and 14. In certain embodiments, the check information may be acquired using the scanner 48 (FIG. 2). For example, the graphical element 336 may be selected to scan a bar code on the check using the scan device 48 (FIG. 2). In other embodiments, the payment information may be acquired by scanning an NFC enabled check. In these embodiments, a graphical element may be included within the summary bar 152 to enter the check information using near field communication. The check information may be acquired by near field communication using techniques similar to those described above with respect to FIGS. 11-12."); [0162] ("In certain embodiments, the approval step may include transmitting the check information to a verification service for approval. For example, the merchant may hire a third party check verification service to verify checks received from customers. In these embodiments, selection of the graphical element 228 may cause the device to send the check information to the external server 96 (FIG. 4), which may in turn send the check information to the check verification service. The check verification service may approve the check for payment and transmit the approval to the electronic device 10 through the external server. Upon receipt, the payment authorization may be displayed within the window 242. The check information also may be transmitted to the customer's bank to verify that the customer has sufficient funds to process the transaction. Further, in other embodiments, the approval process may simply include verifying that all the check information has been entered, and no additional verification may be performed. After the transaction has been approved, the merchant may select the POP graphical element 224 to conduct another transaction.").

- Rodriguez at [0035] ("FIG. 3 depicts an embodiment of a POS terminal 300 to generate and process an e-check transaction. POS terminal 300 includes hardware and software adapted to generate the e-check based upon receipt of a check 305 and to transmit the e-check to a bank for payment based upon the routing number inscribed on check 305. POS terminal 300 may be any type of processor-based device like a desktop computer, a laptop, a PDA, or the like that can be adapted to generate and transmit an e-check. For example, POS terminal 300 may comprise a processor-based cash register at a department store. The cashier that operates the cash register may receive check 305 as payment for some merchandise and insert check 305 into POS terminal 300. Upon generating an e-check, POS terminal 300 may locate an IP address associated with the routing number of the purchaser's account and interact with the purchaser's bank via bank Application Program

Interface (API) 364 to clear check 305. In other embodiments, POS terminal may interface with a second computer system that will facilitate clearance of check 305 by utilizing the corresponding e-check.").

- Nepomniachtchi at 4:65–5:7 (The present invention is directed towards automated document processing and systems and methods for document image processing using mobile devices. Generally, some embodiments of the invention capture an original color image of a document using a mobile device and then converted the color image to a bi-tonal image. More specifically, some embodiments accept a color image of a check taken by a mobile device and convert it into a bi-tonal image of the check that is substantially equivalent in its resolution, size, and quality to check images produced by "standard" scanners); 5:60-6:7 (FIG. 1 is a diagram illustrating an example check 100 that might be imaged with the systems and methods described herein. The mobile image capture and processing systems and methods may be used with a variety of documents, including financial documents such as personal checks, business checks, cashier's checks, certified checks, warrants or other documents. By using an image of the check 100, the check clearing process is performed more efficiently. As would be appreciated by those of skill in the art, checks are not the only type of documents that may be used with these systems. For example, other documents, such as deposit slips, might also be processed using the systems and methods described herein. FIG. 2 is a diagram illustrating an example payment coupon 200 that might be imaged using the systems and methods described herein.); *see also id.* at 9:22–29; 23:11–12; 23:15–16.

- ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone").

- WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a

homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

#### c. Perform OCR on a Check Image to Extract Deposit Information ('310, '095 Patent)

Before the Asserted Patents, it was well known to perform OCR on an image of a check to extract deposit information, including the magnetic ink character recognition ("MICR") line and check amount. For example:

- Cohen at [0022] ("Referring now to FIGS. 1 and 2, at the conclusion of either the scanning or data importation process step 202, in step 204 another process is triggered which performs OCR as well as parsing of the MICR Line information. In a subsequent step 206, the OCR and parsing processes return value data, including: (1) check amount; (2) bank routing number; (3) bank account number; and (4) check number. Significantly, these processes also return quantitative measurements, or values, representing the confidence level of the accuracy of the OCR results pertaining to each of the respective pieces of data. Subsequently, in step 208, a remote operator at the local merchant site 100 reviews the images and data for each check in the batch. Significantly, the efficacy of this process is greatly enhanced by assigning color-coding in accordance with the respective confidence levels. Preferably, green, yellow and red colors are utilized to provide easy visual indication to the operator as to the confidence level attached to each data item. Subsequently, in step 210, and based upon a configuration setting in the profile that governs the particular batch, the operator is required to manually validate only those data items with a certain confidence level. Termed "Examination Rigor", this profile configuration value may be set to require that only low confidence data (i.e. marked red) need be inspected, or that both low and moderate confidence level data (i.e. marked red or yellow) must be inspected, or that all data (i.e. even items marked green) must be inspected for errors. Upon completion of the review, the batch is ready for transmission to the DirectFed™ clearing system 118, or a like third party system. For convenience, all further description herein will refer to transmission to the DirectFed™ clearing system 118. In step 212, the image and data information is packaged at the client 100 for subsequent transmission, in step 214, to the DirectFed™ clearing system 118 via the Internet utilizing an encrypted web service.").

- Acharya at 3:34–47 ("Certain instructions may also be stored in RCT memory 104 and executed by the CPU 103, for example, to assist in communication and data collection functions. For example, optical character recognition software may be used in conjunction with the DIS or the digital camera to convert machine printed characters on the financial instrument or the digital image of the financial instrument to electronic text. Likewise, intelligent character recognition software

may be used to convert handwritten characters on the financial instrument or on the digital image of the financial instrument to electronic text. Additionally, any browser, such as Netscape Navigator™, Internet Explorer™ or Mozilla Firefox™, may be used for Internet communications."); 2:45–67 ("To achieve these and other objects of the invention, a payee would deposit a financial instrument, such as a third party paper check, using a Remote Customer Terminal (RCT) that can communicate with a BOFD system. The payee may prepare, or receive from a third party, a digital image of the financial instrument that is to be deposited. Software and/or hardware accessible at or through the RCT can "recognize" data in or on the digital image of the financial instrument. Such data could include data contained in the MICR line, RTN, account number, amount of financial instrument, date of financial instrument or any other characters present in or on the digital image of the financial instrument. Recognition can be made of such data, whether in human-visible or machine readable form. Alternatively or additionally, the banking customer can input data taken from the financial instrument, or the digital image of the financial instrument, directly into the RCT. In general terms, the RCT may be equipped to "accept" digital images and data regarding the financial instrument. The digital image of the financial instrument and/or data recognized from the financial instrument or inputted into the RCT by the banking customer is transmitted from the RCT to the BOFD system. The BOFD system can process the deposit transaction in several different fashions.").

- Lin at [0155] ("FIG. 14 is a flowchart depicting an exemplary method 310 for acquiring payment information using a camera as illustrated in FIG. 13. The method may begin by receiving (block 312) a card image. The image may be received by taking a picture of the card using the camera 46 (FIG. 2) of the electronic device. The electronic device may translate (block 314) the captured image into the payment information. The translation may be performed using optical character recognition (OCR) or other vision based recognition techniques. In certain embodiments, software for performing OCR may be stored within the storage 54 (FIG. 3) of the electronic device."); [0162] ("In certain embodiments, the approval step may include transmitting the check information to a verification service for approval. For example, the merchant may hire a third party check verification service to verify checks received from customers. In these embodiments, selection of the graphical element 228 may cause the device to send the check information to the external server 96 (FIG. 4), which may in turn send the check information to the check verification service. The check verification service may approve the check for payment and transmit the approval to the electronic device 10 through the external server. Upon receipt, the payment authorization may be displayed within the window 242. The check information also may be transmitted to the customer's bank to verify that the customer has sufficient funds to process the transaction. Further, in other embodiments, the approval process may simply include verifying that all the check information has been entered, and no additional verification may be performed. After the transaction has been approved, the merchant may select the POP graphical element 224 to conduct another transaction.").

- Rodriguez at [0037] ("Character scanner 320 may determine the routing number, account number and, possibly, the check number on check 305. More specifically, character scanner 320 may include magnetic sensors coupled with a MICR 322 to read magnetic ink used to print the routing number, account number, and check number on the check 305. In some embodiments, character scanner 320 may include logic that couples with the optical sensors of image scanner 310 such as optical character recognition 324 to determine the routing number, account number, and check number on the check 305. In such embodiments, optical character recognition 324 may also read the numbers from the check to determine the amount of currency represented by check 305.").

- Fellerman at [0042] ("The check scanner 28 is understood not to segment the captured data into various blocks described above. However, it is contemplated that the bank routing number 40 and the account identifier 42, which are inscribed as MICR data, may be read and recognized in addition to the captured raw image or raster data of the check 24. The MICR data may be appended to the image of the check 24, or may be transmitted separately. The data is then combined with additional information provided to the remote terminal 26 for transmission to a central payment processor 52, the details of which will be described more fully below.").

- Viera at 7:1-32 ("FIG. 4 illustrates a method of performing a transaction on a network according to one embodiment of the present invention. FIG. 4 illustrates a method of performing a deposit transaction. At step 401, a user inserts a check into an image scanner at a kiosk. At step 402, the back and front of the check are scanned and the images are stored (e.g. bitmaps). The magnetic encoding ("MICR") at the bottom of the check is also scanned. The user enters the amount of the deposit using a touchscreen keypad at step 403. At step 404, the bitmap images and MICR obtained from step 402 are passed to an optical character recognition ("OCR") program. OCR will perform several functions. It will re-read the MICR using optical instead of magnetic techniques. It will detect the check amount, payee name, and check number. It will detect the imprinted name of the paying financial institution. It will determine if a signature is present on the front of the check. It will also determine if a signature is present in the endorsement section on the back of the check. A deposit type is determined at step 405 using the information obtained from step 404. In making this determination, risk policies and routing group information from database tables 406 and 407 are used. If deposit is rejected at step 408, the check is returned to the user. If the deposit is accepted, it will be examined for hold override policies using policy information stored in database 411. Funds availability is calculated and presented to the user at step 412. If the user rejects the deposit at step 413, then the check will be returned at step 414. If the user elects to proceed, deposit information will be printed on the check and stored in a secure internal bin at step 415. At step 416, the check images and deposit information will be posted through the shared branching network 417 to be received by the appropriate financial institution.").

- Garcia at 11:7-19 ("For its part, the institution that renders the service must have certain electronic means of recognizing, verifying and processing the information sent by the user. These means consist basically of computer equipment connected to the Internet that has an optical character recognition (OCR) system, a code authentication system and means of transmitting data to the institution's document processing system. The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems)").

- Byrne at [0174] ("An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic."); [0170] ("The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item").

- Nepomniachtchi at 9:53–64 ("In some embodiments, the system might determine the amount by processing the image. For example, in some cases, optical character recognition ("OCR") might be used to determine what characters and numbers are present on the document. For example, numbers located in the amount box of a check or payment coupon might then be determined using OCR or other computer based character determination. This might be done instead of requiring the amount to be entered manually. In other embodiments, a manual entry might be used to verify a computer generated value that is determined using, for example, OCR or other computer based character determination.")

- CAM (Parikh 2005b) at Abstract ("During our work with community microfinance groups in rural India, we found that paper plays a crucial role in many local information practices. However, paper-based record-keeping can be inefficient, so we need to link paper with the flexibility of modern information tools. A mobile phone can be the perfect bridging device. Here we present the CAM mobile document processing system, in which a camera phone is used as an image capture and data entry device. Our system is able to process paper forms containing CamShell programs - embedded instructions that are decoded from an electronic image. By combining 1) paper, 2) audio, 3) numeric data entry, 4) narrative scripted execution and 5) asynchronous connectivity, we have synthesized our experience into a system that we believe is well-suited for an important set of users and applications in the developing world."); Pervasive Computing at 76 ("The CamBrowser application's primary purpose is to capture a visual image of the form and decode and interpret the embedded CamShell program, which executes the user interface that collects information from the user. CamBrowser executes instructions in sequence, starting with instruction zero. Internally, a simple virtual machine keeps track of the current program counter and local variables. You can scan codes either by pointing the camera at the CamForm (scan mode) or by explicitly taking a picture (click mode). We've found the former to be more suitable for quick, directed user interaction, while the

latter is more effective for capturing an entire CamShell program and entering large amounts of data—for example, in a loop.“).

- ImageNet, J&B Software Selects Mitek at 1 (“J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones”), J&B to Showcase Mobile Deposit at 1 (“J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference”); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 (“NCR unleashes remote check deposit by using mobile phone camera”); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 (“RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone”); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 (“Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application”); Your iPhone will be your wallet - 9to5Mac at Page 1 (“We’ve mused on this before, but things have taken another step forward with news of Mitek’s Mobile Deposit (read, banking) application for the iPhone”).

- WVUFCU (iPhoneApp) at 1 (“iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone”); Finovate Blog at 2 (“But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner.”); An Eye for Technology at 1 (“With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.”).

#### d. Validate Information Read From Check Image (’310, ’095 Patent)

Before the Asserted Patents, it was well known to validate information from a check image, including by evaluating the accuracy of the information or comparing information read from the check itself other available information. For example:

- Cohen at [0022] (“Referring now to FIGS. 1 and 2, at the conclusion of either the scanning or data importation process step 202, in step 204 another process is triggered which performs OCR as well as parsing of the MICR Line information. In a subsequent step 206, the OCR and parsing processes return value data, including: (1) check amount; (2) bank routing number; (3) bank account number; and (4) check number. Significantly, these processes also return quantitative measurements, or values, representing the confidence level of the accuracy of the OCR results pertaining to each of the respective pieces of data. Subsequently, in

step 208, a remote operator at the local merchant site 100 reviews the images and data for each check in the batch. Significantly, the efficacy of this process is greatly enhanced by assigning color-coding in accordance with the respective confidence levels. Preferably, green, yellow and red colors are utilized to provide easy visual indication to the operator as to the confidence level attached to each data item. Subsequently, in step 210, and based upon a configuration setting in the profile that governs the particular batch, the operator is required to manually validate only those data items with a certain confidence level. Termed "Examination Rigor", this profile configuration value may be set to require that only low confidence data (i.e. marked red) need be inspected, or that both low and moderate confidence level data (i.e. marked red or yellow) must be inspected, or that all data (i.e. even items marked green) must be inspected for errors. Upon completion of the review, the batch is ready for transmission to the DirectFed™ clearing system 118, or a like third party system. For convenience, all further description herein will refer to transmission to the DirectFed™ clearing system 118. In step 212, the image and data information is packaged at the client 100 for subsequent transmission, in step 214, to the DirectFed™ clearing system 118 via the Internet utilizing an encrypted web service.").

- Acharya at 2:45–67 ("To achieve these and other objects of the invention, a payee would deposit a financial instrument, such as a third party paper check, using a Remote Customer Terminal (RCT) that can communicate with a BOFD system. The payee may prepare, or receive from a third party, a digital image of the financial instrument that is to be deposited. Software and/or hardware accessible at or through the RCT can "recognize" data in or on the digital image of the financial instrument. Such data could include data contained in the MICR line, RTN, account number, amount of financial instrument, date of financial instrument or any other characters present in or on the digital image of the financial instrument. Recognition can be made of such data, whether in human-visible or machine readable form. Alternatively or additionally, the banking customer can input data taken from the financial instrument, or the digital image of the financial instrument, directly into the RCT. In general terms, the RCT may be equipped to "accept" digital images and data regarding the financial instrument. The digital image of the financial instrument and/or data recognized from the financial instrument or inputted into the RCT by the banking customer is transmitted from the RCT to the BOFD system. The BOFD system can process the deposit transaction in several different fashions.").

- Lin at [0155] ("FIG. 14 is a flowchart depicting an exemplary method 310 for acquiring payment information using a camera as illustrated in FIG. 13. The method may begin by receiving (block 312) a card image. The image may be received by taking a picture of the card using the camera 46 (FIG. 2) of the electronic device. The electronic device may translate (block 314) the captured image into the payment information. The translation may be performed using optical character recognition (OCR) or other vision based recognition techniques. In certain embodiments, software for performing OCR may be stored within the storage 54 (FIG. 3) of the electronic device."); [0162] ("In certain embodiments,

the approval step may include transmitting the check information to a verification service for approval. For example, the merchant may hire a third party check verification service to verify checks received from customers. In these embodiments, selection of the graphical element 228 may cause the device to send the check information to the external server 96 (FIG. 4), which may in turn send the check information to the check verification service. The check verification service may approve the check for payment and transmit the approval to the electronic device 10 through the external server. Upon receipt, the payment authorization may be displayed within the window 242. The check information also may be transmitted to the customer's bank to verify that the customer has sufficient funds to process the transaction. Further, in other embodiments, the approval process may simply include verifying that all the check information has been entered, and no additional verification may be performed. After the transaction has been approved, the merchant may select the POP graphical element 224 to conduct another transaction.").

- Rodriguez at [0043] ("Once the cashier receives payment by check and verifies that the amount is correct, i.e. is consistent with the amount shown on the POS terminal, the check is scanned and a check object (e-check) is created electronically at the POS terminal. The POS terminal may be enhanced to allow scanning an image of the check in addition to reading the encoded (MICR) information on the check, identifying the check information (element 420). The POS terminal software is also enhanced to generate an electronic check object, or e-check (element 425), that contains as an attachment the true image copy of the check, and the data entered by the cashier. Thus, the physical check document does not need to be further processed by other parties and may be defaced and returned to the purchaser or forwarded to the purchaser's bank."); [0044] ("In some embodiments, the physical check may be held by the merchant for a predetermined period of time, or may be sent to the merchant's bank for routing to the purchaser's bank. If the check is to be returned to the purchaser, the amount of the check can be retrieved from the e-check and printed on the check. The ink utilized to the print the amount on the check wouldn't have to be MICR at this point because the amount is only printed on the paper check to let the purchaser compare the handwritten amount with the amount actually debited from the purchaser's account. Optionally, the amount of the check in the check object may be added to the scanned check image so that a hard copy of the image can be used as proof of payment.").

- Fellerman at [0046] ("Next, in step 512 the check number 44 from the first of the plurality of checks to be scanned is entered via the keypad 54, and in step 512, the amount specified thereby is likewise entered via the keypad 54. Again, the screen 54 displays prompts corresponding to the respective data inputs. Then, the check scanner 28 is activated and the check 24 is fed for scanning in step 516. After scanning the front and back sides of the check 24, basic verifications are performed according to decision block 518 to ensure that the image data was properly captured. For example, if the check scanner 28 timed out before receiving any data, then an indicator would be activated to prompt a re-scan.

Although the check number and check amount data entered into the remote terminal 26 via the keypad 54 corresponds to the scanned check images, the check images are understood to be stored independently in certain embodiments."), [0048] ("All of the entered data is then verified in a check summary confirmation step 524 and a detail confirmation step 526. In one embodiment, the total of all the checks 24 presented as keyed into the remote terminal 26 in step 514 may confirmed. However, other related aggregate or summary data relating to the checks 24 may also be confirmed. Those having ordinary skill in the art will readily devise such alternative summary data. In the detail confirmation step 526, all or part of the entered data, such as the identification number, date of birth, issuing state, phone number, the check number for each of the scanned checks, and so forth may be verified.").

- Viera at 7:1-32 ("FIG. 4 illustrates a method of performing a transaction on a network according to one embodiment of the present invention. FIG. 4 illustrates a method of performing a deposit transaction. At step 401, a user inserts a check into an image scanner at a kiosk. At step 402, the back and front of the check are scanned and the images are stored (e.g. bitmaps). The magnetic encoding ("MICR") at the bottom of the check is also scanned. The user enters the amount of the deposit using a touchscreen keypad at step 403. At step 404, the bitmap images and MICR obtained from step 402 are passed to an optical character recognition ("OCR") program. OCR will perform several functions. It will re-read the MICR using optical instead of magnetic techniques. It will detect the check amount, payee name, and check number. It will detect the imprinted name of the paying financial institution. It will determine if a signature is present on the front of the check. It will also determine if a signature is present in the endorsement section on the back of the check. A deposit type is determined at step 405 using the information obtained from step 404. In making this determination, risk policies and routing group information from database tables 406 and 407 are used. If deposit is rejected at step 408, the check is returned to the user. If the deposit is accepted, it will be examined for hold override policies using policy information stored in database 411. Funds availability is calculated and presented to the user at step 412. If the user rejects the deposit at step 413, then the check will be returned at step 414. If the user elects to proceed, deposit information will be printed on the check and stored in a secure internal bin at step 415. At step 416, the check images and deposit information will be posted through the shared branching network 417 to be received by the appropriate financial institution."), 7:32-64 ("FIGS. 5A-E are example graphical user interfaces illustrating a deposit transaction according to another embodiment of the present invention. FIG. 5A illustrates a menu of services 501 that is presented to the user. The user will select a service by touching a labeled button 502. If the user wants to end the transaction, he or she can touch the exit button 503. FIG. 5B illustrates a listing of the user's accounts 504. Each account is displayed with the account type 505, the balance 506, and the available funds 507. The user selects the account that will be receiving the deposit. FIG. 5C illustrates a check deposit 508. A check is inserted into an image scanner without an envelope. The account type 509 is displayed. Verification of check scan 510 is displayed. The user uses a keypad 511 to enter

the deposit amount 512. An image 513 of the check is displayed to the user. FIG. 5D illustrates an appraisal of funds availability. The account type 515 is displayed. The deposit amount 516 is also displayed. The funds that are immediately available are displayed at 517A. The availability of the remaining funds is displayed at 517B. The user can reject the deposit by touching button 519 or button 518 to accept. If the deposit is accepted, the deposit information is stored in a remote database that may be accessed by the user's financial institution. For example, after the transaction has completed, FIG. 5E illustrates instant online review of the deposit. A financial institution employee can review the details of the deposit and also see an image of the actual check. Previously, a check must be placed into an envelope for deposit. The envelope then sits in a depository until a teller can process the deposit and determine the availability. A user can then find out what the funds availability is through contacting the financial institution. The institution may also send a letter to the account holder's home with the funds availability.").

- Garcia at 11:7-19 ("For its part, the institution that renders the service must have certain electronic means of recognizing, verifying and processing the information sent by the user. These means consist basically of computer equipment connected to the Internet that has an optical character recognition (OCR) system, a code authentication system and means of transmitting data to the institution's document processing system. The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems)").

- Byrne at [0058] ("An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the

"yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.").

- Nepomniachtchi at Abstract ("The present invention relates to automated document processing and more particularly, to methods and systems for document image capture and processing using mobile devices. In accordance with various embodiments, methods and systems for document image capture on a mobile communication device are provided such that the image is optimized and enhanced for data extraction from the document as depicted. These methods and systems may comprise capturing an image of a document using a mobile communication device; submitting the image to a server; and processing the image to create a bi-tonal image of the document for data extraction. Additionally, these methods and systems may comprise capturing a first image of a document using the mobile communication device; automatically detecting the document within the image; geometrically correcting the image; binarizing the image; correcting the orientation of the image; correcting the size of the image; and outputting the resulting image of the document."); *see also* 1:12-16, 48-55, 1:56-2:3; 2:4-12; 2:59-65; 4:49-57; 4:65-5:7; 5:60-6:7; 6:8-26; 6:27-45; 9:22-29; 9:53-64; 16:32-52; 18:62–19:18; 19:51–67; 23:11-16; claim 7.

- ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone").

- WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a

native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

**e. Confirming Check Deposit Can Go Forward ('310, '095 Patent)**

Before the Asserted Patents, it was well known to confirm that check deposit can go forward, including by confirming accuracy of information read from the check in software and by asking a user to confirm that they wish to proceed with deposit. For example:

- Cohen at [0022] ("Referring now to FIGS. 1 and 2, at the conclusion of either the scanning or data importation process step 202, in step 204 another process is triggered which performs OCR as well as parsing of the MICR Line information. In a subsequent step 206, the OCR and parsing processes return value data, including: (1) check amount; (2) bank routing number; (3) bank account number; and (4) check number. Significantly, these processes also return quantitative measurements, or values, representing the confidence level of the accuracy of the OCR results pertaining to each of the respective pieces of data. Subsequently, in step 208, a remote operator at the local merchant site 100 reviews the images and data for each check in the batch. Significantly, the efficacy of this process is greatly enhanced by assigning color-coding in accordance with the respective confidence levels. Preferably, green, yellow and red colors are utilized to provide easy visual indication to the operator as to the confidence level attached to each data item. Subsequently, in step 210, and based upon a configuration setting in the profile that governs the particular batch, the operator is required to manually validate only those data items with a certain confidence level. Termed "Examination Rigor", this profile configuration value may be set to require that only low confidence data (i.e. marked red) need be inspected, or that both low and moderate confidence level data (i.e. marked red or yellow) must be inspected, or that all data (i.e. even items marked green) must be inspected for errors. Upon completion of the review, the batch is ready for transmission to the DirectFed™ clearing system 118, or a like third party system. For convenience, all further description herein will refer to transmission to the DirectFed™ clearing system 118. In step 212, the image and data information is packaged at the client 100 for subsequent transmission, in step 214, to the DirectFed™ clearing system 118 via the Internet utilizing an encrypted web service.").

- Acharya at 2:45–67 ("To achieve these and other objects of the invention, a payee would deposit a financial instrument, such as a third party paper check, using a Remote Customer Terminal (RCT) that can communicate with a BOFD system. The payee may prepare, or receive from a third party, a digital image of the

financial instrument that is to be deposited. Software and/or hardware accessible at or through the RCT can "recognize" data in or on the digital image of the financial instrument. Such data could include data contained in the MICR line, RTN, account number, amount of financial instrument, date of financial instrument or any other characters present in or on the digital image of the financial instrument. Recognition can be made of such data, whether in human-visible or machine readable form. Alternatively or additionally, the banking customer can input data taken from the financial instrument, or the digital image of the financial instrument, directly into the RCT. In general terms, the RCT may be equipped to "accept" digital images and data regarding the financial instrument. The digital image of the financial instrument and/or data recognized from the financial instrument or inputted into the RCT by the banking customer is transmitted from the RCT to the BOFD system. The BOFD system can process the deposit transaction in several different fashions.")

- Lin at [0161] ("The screen also includes graphical elements 334 and 228 for navigating between screens. Specifically, the graphical element 334 allows the merchant to return to the payment screen 198 to receive a different form of payment. Once the check information has been entered, using the keypad 266 (FIG. 11), the camera 46 (FIG. 2), the scanner 48 (FIG. 2), or the NFC device 44 (FIG. 1), the merchant may select the graphical element 228 to process the payment and advance to the payment screen 240. As noted above, the payment screen 240 includes the display window 242 indicating the status of the transaction.").

- Rodriguez at [0043] ("Once the cashier receives payment by check and verifies that the amount is correct, i.e. is consistent with the amount shown on the POS terminal, the check is scanned and a check object (e-check) is created electronically at the POS terminal. The POS terminal may be enhanced to allow scanning an image of the check in addition to reading the encoded (MICR) information on the check, identifying the check information (element 420). The POS terminal software is also enhanced to generate an electronic check object, or e-check (element 425), that contains as an attachment the true image copy of the check, and the data entered by the cashier. Thus, the physical check document does not need to be further processed by other parties and may be defaced and returned to the purchaser or forwarded to the purchaser's bank."); [0044] ("In some embodiments, the physical check may be held by the merchant for a predetermined period of time, or may be sent to the merchant's bank for routing to the purchaser's bank. If the check is to be returned to the purchaser, the amount of the check can be retrieved from the e-check and printed on the check. The ink utilized to the print the amount on the check wouldn't have to be MICR at this point because the amount is only printed on the paper check to let the purchaser compare the handwritten amount with the amount actually debited from the purchaser's account. Optionally, the amount of the check in the check object may be added to the scanned check image so that a hard copy of the image can be used as proof of payment."); [0033] ("Purchaser's bank 140 may include a bank or a computer system for a bank to process debits from purchaser's account 144. In

particular, purchaser's bank 140 may include e-check clearer 142 and purchaser's account 144. Upon receiving an e-check from merchant 120 based upon a transaction involving check 115, e-check clearer 142 may verify the account number of purchaser's account 144 and compare the amount 218 to the balance of purchaser's account 144 to determine whether the transaction should be approved. In further embodiments, the check number may be compared with the check number of previously processed checks to verify that the same check 115 is not being processed again. In one embodiment, when additional information such as purchaser's driver license number, name, PIN and/or other identifying information is included in the e-check 200, e-check clearer 142 may compare that information with information associated with purchaser's account 144 to confirm authorization to access funds in the account.").

- Fellerman at [0046] ("Next, in step 512 the check number 44 from the first of the plurality of checks to be scanned is entered via the keypad 54, and in step 512, the amount specified thereby is likewise entered via the keypad 54. Again, the screen 54 displays prompts corresponding to the respective data inputs. Then, the check scanner 28 is activated and the check 24 is fed for scanning in step 516. After scanning the front and back sides of the check 24, basic verifications are performed according to decision block 518 to ensure that the image data was properly captured. For example, if the check scanner 28 timed out before receiving any data, then an indicator would be activated to prompt a re-scan. Although the check number and check amount data entered into the remote terminal 26 via the keypad 54 corresponds to the scanned check images, the check images are understood to be stored independently in certain embodiments."), [0048] ("All of the entered data is then verified in a check summary confirmation step 524 and a detail confirmation step 526. In one embodiment, the total of all the checks 24 presented as keyed into the remote terminal 26 in step 514 may confirmed. However, other related aggregate or summary data relating to the checks 24 may also be confirmed. Those having ordinary skill in the art will readily devise such alternative summary data. In the detail confirmation step 526, all or part of the entered data, such as the identification number, date of birth, issuing state, phone number, the check number for each of the scanned checks, and so forth may be verified.").

- Viera at 7:1-32 ("FIG. 4 illustrates a method of performing a transaction on a network according to one embodiment of the present invention. FIG. 4 illustrates a method of performing a deposit transaction. At step 401, a user inserts a check into an image scanner at a kiosk. At step 402, the back and front of the check are scanned and the images are stored (e.g. bitmaps). The magnetic encoding ("MICR") at the bottom of the check is also scanned. The user enters the amount of the deposit using a touchscreen keypad at step 403. At step 404, the bitmap images and MICR obtained from step 402 are passed to an optical character recognition ("OCR") program. OCR will perform several functions. It will re-read the MICR using optical instead of magnetic techniques. It will detect the check amount, payee name, and check number. It will detect the imprinted name of the paying financial institution. It will determine if a signature is present on the front

of the check. It will also determine if a signature is present in the endorsement section on the back of the check. A deposit type is determined at step 405 using the information obtained from step 404. In making this determination, risk policies and routing group information from database tables 406 and 407 are used. If deposit is rejected at step 408, the check is returned to the user. If the deposit is accepted, it will be examined for hold override policies using policy information stored in database 411. Funds availability is calculated and presented to the user at step 412. If the user rejects the deposit at step 413, then the check will be returned at step 414. If the user elects to proceed, deposit information will be printed on the check and stored in a secure internal bin at step 415. At step 416, the check images and deposit information will be posted through the shared branching network 417 to be received by the appropriate financial institution."), 7:32-64 ("FIGS. 5A-E are example graphical user interfaces illustrating a deposit transaction according to another embodiment of the present invention. FIG. 5A illustrates a menu of services 501 that is presented to the user. The user will select a service by touching a labeled button 502. If the user wants to end the transaction, he or she can touch the exit button 503. FIG. 5B illustrates a listing of the user's accounts 504. Each account is displayed with the account type 505, the balance 506, and the available funds 507. The user selects the account that will be receiving the deposit. FIG. 5C illustrates a check deposit 508. A check is inserted into an image scanner without an envelope. The account type 509 is displayed. Verification of check scan 510 is displayed. The user uses a keypad 511 to enter the deposit amount 512. An image 513 of the check is displayed to the user. FIG. 5D illustrates an appraisal of funds availability. The account type 515 is displayed. The deposit amount 516 is also displayed. The funds that are immediately available are displayed at 517A. The availability of the remaining funds is displayed at 517B. The user can reject the deposit by touching button 519 or button 518 to accept. If the deposit is accepted, the deposit information is stored in a remote database that may be accessed by the user's financial institution. For example, after the transaction has completed, FIG. 5E illustrates instant online review of the deposit. A financial institution employee can review the details of the deposit and also see an image of the actual check. Previously, a check must be placed into an envelope for deposit. The envelope then sits in a depository until a teller can process the deposit and determine the availability. A user can then find out what the funds availability is through contacting the financial institution. The institution may also send a letter to the account holder's home with the funds availability.").

- Byrne at [0167] ("Deposit bag is a holding/receiving area until the customer commits the deposit. Items are not committed to ICING until the bag is closed;"); [0173] ("Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g., from a dropdown list. The system creates a bag for this session 602.); [0191] ("The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the

same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710."); *see also* FIGs. 6 and 7; [0058] ("The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending Such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition."); [0191] ("The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.").

- Nepomniachtchi at 9:22–29 ("In operation 702, in the illustrated embodiment, the type of document is selected. For example, a user might select a document type for a check, payment coupon or deposit slip. By entering the type of document, a mobile device might be able to scan specific parts of an image to determine, for example, payee, check amount, signature, etc. In some embodiments, however, a device might determine what type of document is being imaged by processing the image"); 9:53–64 ("In some embodiments, the system might determine the amount by processing the image. For example, in some cases, optical character recognition ("OCR") might be used to determine what characters and numbers are present on the document. For example, numbers located in the amount box of a check or payment coupon might then be determined using OCR or other computer based character determination. This might be done instead of requiring the amount to be entered manually. In other embodiments, a manual entry might be used to verify a computer generated value that is determined using, for example, OCR or other computer based character determination.")

- ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI

TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone").

- WVUFCU Finovate Blog at 2 ("Users take a picture by pressing the button on the left(see inset), verifying that the image is readable, then uploading with the button on the right."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that."); iPhoneApp at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone … Step 2: Resize or adjust your image Step 3: Touch "iDeposit" to send your image").

**f. Downloaded Applications ('310, '095 Patent)**

Before the Asserted Patents, it was well known to download applications, including downloading third party applications on mobile devices that could control the devices' cameras. For example:

- Meier at 5 ("The Android Marketplace client for downloading thied-party [*sic*] Android applications.").

- Goyal at 66 ("Of course, simply requesting permissions in the JAD file is only half the story. Any rogue MIDlet could request the permission this way and cause trouble. You still need to establish some sort of trust principle between the MIDlet and the AMS; the AMS will then trust your MIDlet enough to let it be installed in a trusted domain. This is done by signing the MIDlet with a certificate that can be validated up to a root certificate installed on the device.… After the MIDlet is signed, you can test it by using the Run via OTA menu option shown in Figure 5-6. You don't use the Run menu option like you have been doing so far because when you run the MIDlet locally (by selecting the Run menu option), it does not get installed into a protection domain. Only when installed via the Run via OTA option does your MIDlet gets a chance to be installed into a requested domain, which is the domain that you request by specifying permission for it in the JAD file.").

- Acharya at 3:1–20 (“In one embodiment of the invention, the RCT is located at home, and comprises a digital camera, or a personal computer or wireless handheld device, such as a Blackberry® or Palm Treo®, having digital imaging capabilities such as an attached image scanner or a digital camera. To initiate the transaction, the banking customer uses an Internet-based application loaded on the RCT to access the BOFD system. A pre-assigned Personal Identification Number (PIN) or other password or secret, including a biometric, may be required to allow the banking customer access to the BOFD system. Either when prompted or prior to accessing the BOFD system, the banking customer captures the digital image of the financial instrument by the scanner or the digital camera and prepares a file storing the digital image. Upon prompting by the BOFD system, the banking customer forwards the digital image to the BOFD system along with the data that was either ‘recognized’ from the digital image of the financial instrument or was input into the RCT by the banking customer. Alternatively, the customer may input data regarding the financial instrument directly into the BOFD system.”).

- Lin at [0070] (“When an icon 30 is selected, the device 10 may be configured to open an application associated with that icon and display a corresponding screen. For example, when the Point of Purchase (POP) icon 32 is selected, the device 10 may be configured to open an application for conducting a sales transaction. The application may facilitate sales based transactions such as scanning an article of merchandise, receiving payment information, and retrieving customer information. For each application, screens including additional icons or other GUI elements may be displayed on the display 24. Of course, if the device 10 is configured to be used solely or primarily for sales transactions, the POP icon 32 may not be present, as the related application may already be loaded and running at device startup.”).

- Rodriguez at [0048] (“One embodiment of the invention is implemented as a program product for use with a computer system such as, for example, the system 100 shown in FIG. 1. The program(s) of the program product defines functions of the embodiments (including the methods described herein) and can be contained on a variety of signal-bearing media. Illustrative signal-bearing media include, but are not limited to: (i) information permanently stored on non-writable storage media (e.g., read-only memory devices within a computer such as CD-ROM disks readable by a CD-ROM drive); (ii) alterable information stored on writable storage media (e.g., hard-disk drive or floppy disk within a diskette drive); and (iii) information conveyed to a computer by a communications medium, such as through a computer or telephone network, including wireless communications. The latter embodiment specifically includes information downloaded from the Internet and other networks. Such signal-bearing media, when carrying computer-readable instructions that direct the functions of the present invention, represent embodiments of the present invention.”).

- Fellerman at [0028] (“FIG. 5 is a flowchart of a terminal device software application implementing the electronic deferred check payment issuance system;”).

- Cohen at [0020] ("Referring initially to FIG. 1, a system 100, alternatively referred to herein as the DirectFed™ client system, is provided at a remote merchant location or site. The DirectFed™ client system 100, whose output files have received approval by the Federal Reserve Bank, operates in the following manner. DirectFed™ client software runs locally at a remote merchant location, either on a personal computer (PC) 108 acting as a stand alone workstation or, alternatively, on a networked PC 112, preferably running either a Microsoft Windows XP or Windows 2000 operating system. The DirectFed™ client software, run either on a standalone PC 108 or on a client's network 112, utilizes a common database and image repository 114. Accordingly, participating merchants maintain local archiving of the images, in contrast to conventional centralized systems, which typically maintain central system database archives. The network version allows for a sophisticated workflow design, whereby each PC on the network can be configured in such a way so as to perform all tasks associated with the DirectFed™ client software, or a subset of those tasks. The standalone remote computer 108, or at least one computer in the network environment 112, is equipped with, or is communicatively linked to, a respective scanner 106, 110. For convenience, all further descriptions and examples provided herein will reference a standalone computer.").

- Ranjan at 2 ("The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too.").

- Asai at [0025] ("JAM downloads JAR file and ADF (Application Descriptor File) to mobile phone 10. JAR file is stored in JAR storage 110 a. Written in ADF is information related to JAR file, such as AppClass key specifying the main class name used when activating the corresponding Java-AP or Spsize indicating the capacity of scratch pad 110 b allocated to the Java-AP. JAM refers to the descriptive content of the ADF and manages the execution and the behavior of Java-AP corresponding to the ADF. The management in executing Java-AP involves activation and shut down of Java-AP. The management in behavior of Java-AP includes the bridging of Java-AP and native application program and access control to data stored in non-volatile memory 110."), [0076] ("JAR file including a camera AP and ADF corresponding to the file are assumed to be stored in non-volatile memory 110 after being downloaded to mobile phone 10 from a server device."), [0138] ("(7) In the above embodiment, software having function of camera API is preinstalled in storage unit 107 of mobile phone 10. Software with such a function may also be downloaded from a mobile communications network, or storage medium storing such software may be applied to mobile phone 10.").

- Byrne at Abstract ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); [0003] ("The invention relates to providing a plug-in to drive a device for securely transmitting documents in real time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to Scan checks using a personal computer and deposit the checks via the browser to a bank account."); [0033] ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to Scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); [0041] ("A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); *see also* [0045]-[0049]; FIGs. 1-7.

- Android Developers Website (Android Market) at 1 ("With Android Market, users can easily download apps to their Android-powered phone. Users can also rate the apps they've downloaded and leave comments. These users' ratings along with anonymous usage statistics help determine how apps are ranked and presented within Android Market.").

- Android Website (Open Handset) at 1 ("The Open Handset Alliance, a group of more than 30 technology and mobile companies, is developing Android: the first complete, open, and free mobile platform. To help developers get started developing new applications, we're offering an early look at the Android Software Development Kit. … Download the SDK").

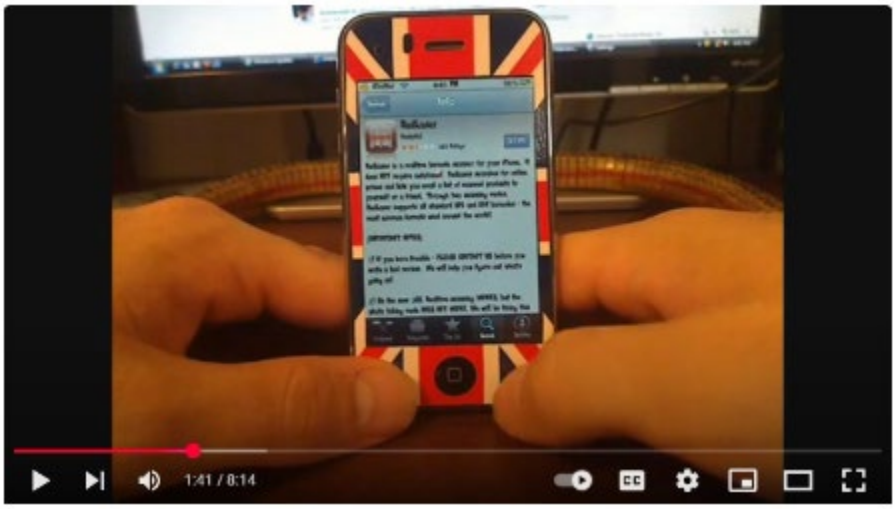
- CAM (Parikh 2007) at 50 ("Developing custom applications for mobile phones is becoming more common. However, this requires knowledge of new APIs (some of which require licensing). Distributing applications is cumbersome. Either providers must push content to users, or users must navigate to web sites and download the software. Due to the difficulty of using the mobile web and the latency in the mobile data network, the easiest way to install an application is by using a PC to download the installer and transfering it to the phone using bluetooth or USB."), 52 ("CAM provides an API for accessing the mobile phone's user interface, networking and multimedia input and output capabilities. Applications are downloaded on demand from an online source, either via the web or a multimedia message (MMS). The XML is cached locally on the phone

along with any associated files, in a directory structure organized according to the data server and application ID.").

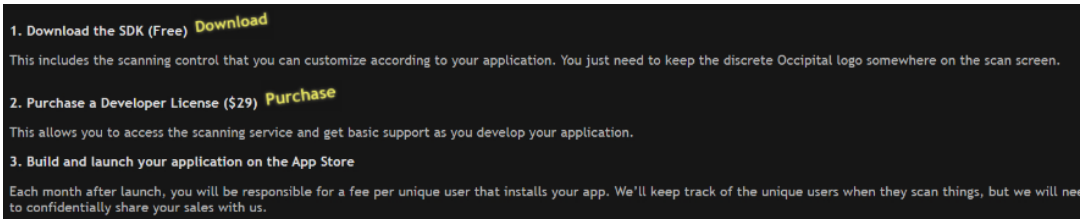
- ImageNet, J&B Software Selects ImageNet at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone").

- WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

- RedLaser (Review Video) at 1:06-3:04



- RedLaser (SDK) at 1




- ScanZoom (Attia) at 3:34-43 ("To utilize the ScanZoom software, a user downloads the ScanZoom onto his/her cell phone or PDA through wireless access protocol ("WAP"), infrared, or Bluetooth® connectivity. However, any protocol which allows a user to download a program to a mobile device may be utilized to download ScanZoom. Once the user has downloaded ScanZoom, the user launches the application. This causes ScanZoom to properly initialize the digital camera coupled to the mobile device to accept input. It starts the digital camera by calling its Application identification.").

### g. Software Abstraction Layers that Control a Device Camera ('310, '095 Patent)

Before the Asserted Patents, it was well known to use a software abstraction layer, such as an application programming interface ("API"), to control a device camera or other image capture device. For example:

- Meier at 315 ("In this chapter, you'll take a closer look at some of these hardware APIs. In particular, you'll learn how to play and record multimedia content including audio, video, and still images, as well as use the camera to capture images and preview and capture live video.").

- Goyal at 174 ("The getSnapshot () method in VideoControl allows you to take a snapshot of the image that is playing on the device screen. Of course, because the getSnapshot () method is a method of the VideoControl class, you can take an image of any video that is playing on a device screen, not just captured video. The method returns the image data as a byte array, which you can then manipulate in any way you want using the LCDUI Image class. … As you may expect by now, the locator to hook into a device camera takes the form of capture://video. This gives you a Player instance that returns video in the default format. The Motorola C975 doesn't support video capture, but the default (and the only) format supported by the BenQ M75 device is video/3gpp.").

- Acharya at 3:1–20 ("In one embodiment of the invention, the RCT is located at home, and comprises a digital camera, or a personal computer or wireless handheld device, such as a Blackberry® or Palm Treo®, having digital imaging capabilities such as an attached image scanner or a digital camera. To initiate the

transaction, the banking customer uses an Internet-based application loaded on the RCT to access the BOFD system. A pre-assigned Personal Identification Number (PIN) or other password or secret, including a biometric, may be required to allow the banking customer access to the BOFD system. Either when prompted or prior to accessing the BOFD system, the banking customer captures the digital image of the financial instrument by the scanner or the digital camera and prepares a file storing the digital image. Upon prompting by the BOFD system, the banking customer forwards the digital image to the BOFD system along with the data that was either 'recognized' from the digital image of the financial instrument or was input into the RCT by the banking customer. Alternatively, the customer may input data regarding the financial instrument directly into the BOFD system.").

- Lin at [0153] ("The payment information also may be acquired from a credit card using a camera instead of near field communication. FIG. 13 illustrates a series of screens for acquiring payment information using the camera 46 shown in FIG. 2. From the card entry screen 218, the merchant may select the camera graphical element 236 to acquire the payment information using the camera. A camera input screen 300 facilitates entry of the payment information using the camera. The screen 300 includes a frame 302 consisting of two corners disposed on the screen 300 to identify the borders of the received image. The frame 302 may function as a viewfinder for the camera. In certain embodiments, the frame 302 may be omitted, and the entire display area may be used to receive the image.").

- Rodriguez at [0035] ("FIG. 3 depicts an embodiment of a POS terminal 300 to generate and process an e-check transaction. POS terminal 300 includes hardware and software adapted to generate the e-check based upon receipt of a check 305 and to transmit the e-check to a bank for payment based upon the routing number inscribed on check 305. POS terminal 300 may be any type of processor-based device like a desktop computer, a laptop, a PDA, or the like that can be adapted to generate and transmit an e-check. For example, POS terminal 300 may comprise a processor-based cash register at a department store. The cashier that operates the cash register may receive check 305 as payment for some merchandise and insert check 305 into POS terminal 300. Upon generating an e-check, POS terminal 300 may locate an IP address associated with the routing number of the purchaser's account and interact with the purchaser's bank via bank Application Program Interface (API) 364 to clear check 305. In other embodiments, POS terminal may interface with a second computer system that will facilitate clearance of check 305 by utilizing the corresponding e-check.").

- Fellerman at [0045] ("For the remote terminal 26 to perform the aforementioned capture, aggregation, and data transmission operations, it is contemplated that the remote terminal 26 is programmed with executable instructions therefor that coordinates the operation of the check scanner 28, the keypad 54, the screen 56, and the communications module 58. Further details of such executable instructions, and in particular those relating to the capture and transmission of the checks 24, are illustrated in the flowchart of FIG. 5. The method begins with a step 500 of receiving an input from the keypad 54 to initiate a deferred payment

transaction. Thereafter, in step 502, an identification card of the purchaser 10 such as a driver's license or other government-issued credential with a magnetic encoding strip is scanned with a magnetic card reader 62. Typical information encoded in a magnetic strip of an identification card includes an identification number, date of birth, and the issuing State, among others. In the absence of a magnetic encoding strip, it is also contemplated that a barcode, if present on the identification card, may be scanned with a bar code reader to capture the same information. If it is determined in the decision block 504 that the read was not successful or that the presented identification card did not have a magnetic strip, the method continues with manual entry steps 506, 507, and 508, where the keypad 54 is used to enter the identification number, a code corresponding to the issuing State, and the date of birth, respectively. Otherwise, such data is read directly from the identification card. Preliminary data collection concludes with a step 510 of entering the phone number of the purchaser 10 via the keypad 54. Throughout each of these data collection steps, the screen 56 displays corresponding prompts therefor to request proper data from the user/merchant 14.").

- Cohen at [0068] ("The Client utilizes several toolkits and processes to perform its tasks. The client can utilize the following commercially available hardware and software: image recognition software such as that developed by A2iA, Inc. of New York, N.Y.; tools for image manipulation such as that commercially available from Lead Technologies of Charlotte, N.C.; scanner control and interface software, such as that commercially available from Silver Bullet, Inc. of Pensacola, Fla.; application development software commercially available from Microsoft Corporation of Redmond, Wash.; Microsoft Web Services and Fileup EE available from SoftArtisans, Inc.; database systems such as SQL Server or Microsoft SQL Server Desktop Edition, commercially available from Microsoft Corporation; and Microsoft Windows XP operating system.").

- Ranjan at 1 ("A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.").

- Asai at [0027] ("Application API includes a camera API. When executing a Java-AP, KVM calls native application with a camera function through a camera API and photographs still image or moving image by camera controlling unit 104."), [0075]-[0076] ("Referring now to FIG. 3, the operation example of CPU 105 will be explained, wherein CPU executes a Java-AP (hereinafter, "camera AP") by compiling a program that is written a method defined by above described camera API and above explained application API as an executable byte code JAR file

including a camera AP and ADF corresponding to the file are assumed to be stored in non-volatile memory 110 after being downloaded to mobile phone 10 from a server device.").

- Android Developers Website (Camera) at 1 ("The Camera class is used to connect/disconnect with the camera service, set capture settings, start/stop preview, snap a picture, and retrieve frames for encoding for video."), 2 ("setPreviewCallback(Camera.PreviewCallback cb) Can be called at any time to instruct the camera to use a callback for each preview frame in addition to displaying it. … takePicture(Camera.ShutterCallbackshutter, Camera.PictureCallback raw, Camera.PictureCallback jpeg) Registers a callback to be invoked when a picture is taken.").

- Android Website (android.hardware.CameraDevice) at 1 ("android.hardware.CameraDevice The CameraDevice class is used to open/release the camera driver, set image parameters and capture a preview frame (320*240 pixels) or a full-resolution image (1280*1024 pixels). This class does not have any public constructors. However, calling the static method open() will return a CameraDevice object. To get a frame/picture, follow these steps: open() the driver. Open will return null if it fails. Set the image parameters (using setCaptureParams(CameraDevice.CaptureParams)). Look at CameraDevice.CaptureParams class for details. Capture (capture(Canvas)) an image with a Canvas object. Note that the Bitmap in the Canvas object must already have enough space allocated for the frame/image. If you are shooting preview, then you would want to call capture(Canvas) several times. It's not necessary to set parameters after each capture(Canvas) if the parameters are the same. Finally, close() the driver.").

- Byrne at [0048] ("The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.").

- Nepomniachtchi at 6:8–16 (“In some embodiments, checks 100, payment coupons 200, or other documents might be imaged using a mobile device. The mobile device may be a mobile telephone handset, Personal Digital Assistant, or other mobile communication device. The mobile device may include a camera, or might include functionality that allows it to connect to a camera. This connection might be wired or wireless. In this way the mobile device may connect to an external camera and receive images from the camera.”).

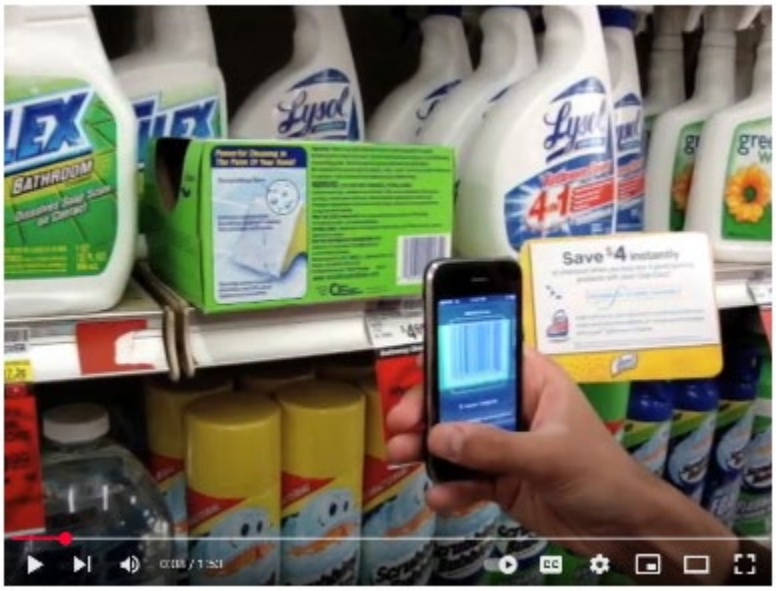
- CAM (Parikh 2005b) at 2 (“CamShell provides an API for accessing the mobile phone’s functionality - including the phone’s user interface, networking and telephony features. Listed below are the main functions currently included in this API:
  • User Dialogs - These instructions launch various user dialogs to collect data, get confirmation or convey a message to the user. Each dialog can be associated with audio and graphical prompts. Audio feedback has been found very useful by potential users.
  • HTTP Post, Get - These instructions are used to make HTTP requests using the phone’s built-in connectivity.
  • SMS, MMS, Email - These instructions transmit asynchronous network messages. In the case of MMS and Email the current form image can be attached.
  • Phone Call - This instruction is used to make a phone call to a particular number.
  • Applications - These instructions are used to launch other applications, such as the Web or WAP browser.”).

- ImageNet, J&B Software Selects Mitek at 1 (“J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones”), J&B to Showcase Mobile Deposit at 1 (“J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference”); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 (“NCR unleashes remote check deposit by using mobile phone camera”); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 (“RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone”); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 (“Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application”); Your iPhone will be your wallet - 9to5Mac at Page 1 (“We’ve mused on this before, but things have taken another step forward with news of Mitek’s Mobile Deposit (read, banking) application for the iPhone”).

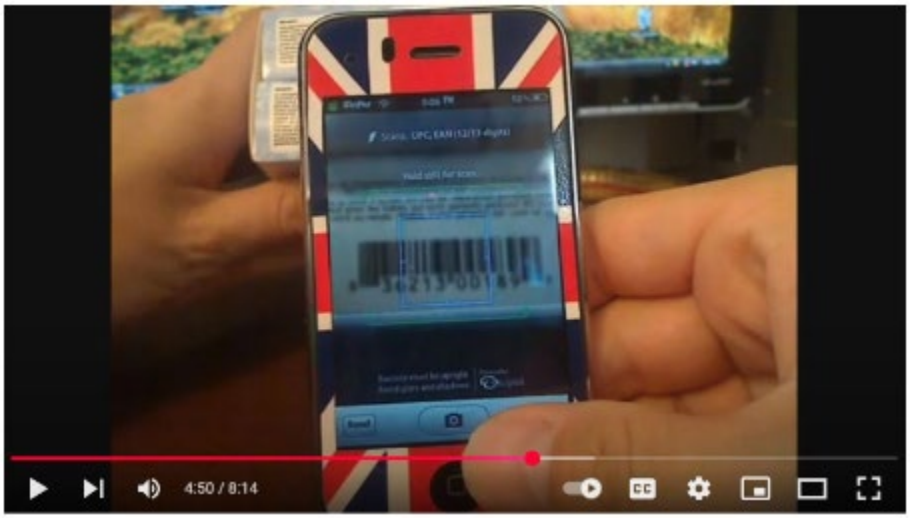
- WVUFCU (iPhoneApp) at 1 (“iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone”); Finovate Blog at 2 (“But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a

homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

- RedLaser (Realtime Video) at 0:01-0:09




- RedLaser (Review Video) at 3:53-5:19




- ScanZoom (Attia) at 7:62-8:3 ("However, if the windows message loop is empty, the user is then directed to push a button on mobile device 105 to take an image of barcode 103 in step 363. Alternatively, the image may be acquired automatically by mobile device 105 for "on-the-fly" decoding. Camera control module 305 interacts directly with API 301 to perform this step since the user must physically

push a button on mobile device 105 to take the image. Alternatively, this feature may be fully automated.").

**h. Mobile Device Cameras Previewing and Capturing an Image from Live Video ('310, '095 Patent)**

Before the Asserted Patents, it was well known to preview and capture images from a live video from a mobile device camera. For example:

- Meier at 320 ("Access to the camera's streaming video means that you can incorporate live video into your applications. Some of the most exciting early Android applications have used this functionality as the basis for augmenting reality. … You'll learn more about Surfaces in the following chapter, although Android includes an excellent example of using a SurfaceView to display the camera preview in real time. This example is available in the graphics/CameraPreview project in the SDK API demos. You can also assign a PreviewCallback to be fi red for each preview frame, allowing you to manipulate or display each preview frame individually. … Take a picture by calling takePicture on a Camera object, passing in a ShutterCallback and PictureCallback implementations for the RAW and JPEG-encoded images. Each picture callback will receive a byte array representing the image in the appropriate format, while the shutter callback is triggered immediately after the shutter is closed.").

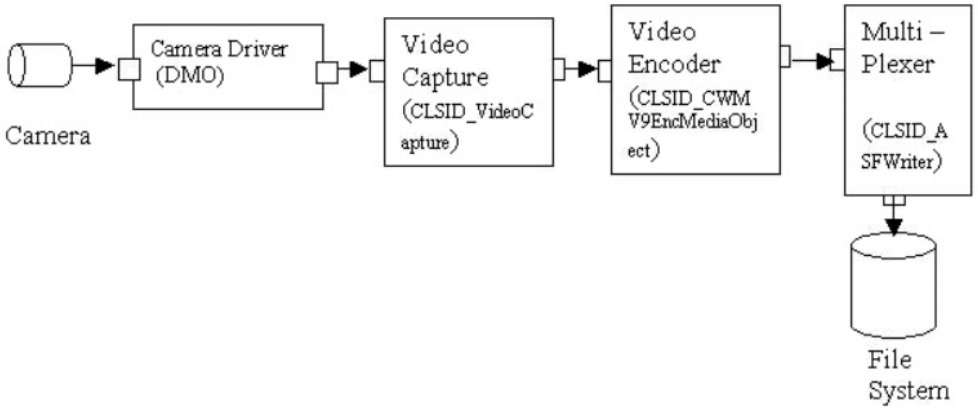
- Goyal at 174 ("The getSnapshot() method in VideoControl allows you to take a snapshot of the image that is playing on the device screen. Of course, because the getSnapshot () method is a method of the VideoControl class, you can take an image of any video that is playing on a device screen, not just captured video. The method returns the image data as a byte array, which you can then manipulate in any way you want using the LCDUI Image class. … Looking Through the Viewfinder As you may expect by now, the locator to hook into a device camera takes the form of capture://video. This gives you a Player instance that returns video in the default format. The Motorola C975 doesn't support video capture, but the default (and the only) format supported by the BenQ M75 device is video/3gpp.").

- Acharya at 3:1–20 ("In one embodiment of the invention, the RCT is located at home, and comprises a digital camera, or a personal computer or wireless handheld device, such as a Blackberry® or Palm Treo®, having digital imaging capabilities such as an attached image scanner or a digital camera. To initiate the transaction, the banking customer uses an Internet-based application loaded on the RCT to access the BOFD system. A pre-assigned Personal Identification Number (PIN) or other password or secret, including a biometric, may be required to allow the banking customer access to the BOFD system. Either when prompted or prior to accessing the BOFD system, the banking customer captures the digital image of the financial instrument by the scanner or the digital camera and prepares a file storing the digital image. Upon prompting by the BOFD system, the banking

customer forwards the digital image to the BOFD system along with the data that was either 'recognized' from the digital image of the financial instrument or was input into the RCT by the banking customer. Alternatively, the customer may input data regarding the financial instrument directly into the BOFD system.").

- Lin at [0153] ("The payment information also may be acquired from a credit card using a camera instead of near field communication. FIG. 13 illustrates a series of screens for acquiring payment information using the camera 46 shown in FIG. 2. From the card entry screen 218, the merchant may select the camera graphical element 236 to acquire the payment information using the camera. A camera input screen 300 facilitates entry of the payment information using the camera. The screen 300 includes a frame 302 consisting of two corners disposed on the screen 300 to identify the borders of the received image. The frame 302 may function as a viewfinder for the camera. In certain embodiments, the frame 302 may be omitted, and the entire display area may be used to receive the image.").

- Ranjan at Fig. 1 ("




").

- Asai at [0017] ("As shown, a camera controlling unit 104 acts as a typical camera that photographs an object. Camera controlling unit 104 includes two types of camera devices; one camera device is for photographing still images; the other camera device is for photographing moving images. Each camera device has a lens, an image pickup device including a CCD (Charge Coupled device), an A/D converter, an image processing circuit, and a shutter. An image of an object is converted into an optical image and formed on an image pickup device. The imaging device converts the intensity of light corresponding to the optical image into quantities of electrical charge. An analogue video signal outputted from an image pickup device is converted by an A/D converter into image data in a digital video signal and displayed on a liquid crystal display 106. Image data photographed by pressing a shutter is stored temporarily in RAM 109. The image processing circuit performs compressing/expanding of image data in accordance with JPEG (Joint Photographic Experts Group image compression technology).").

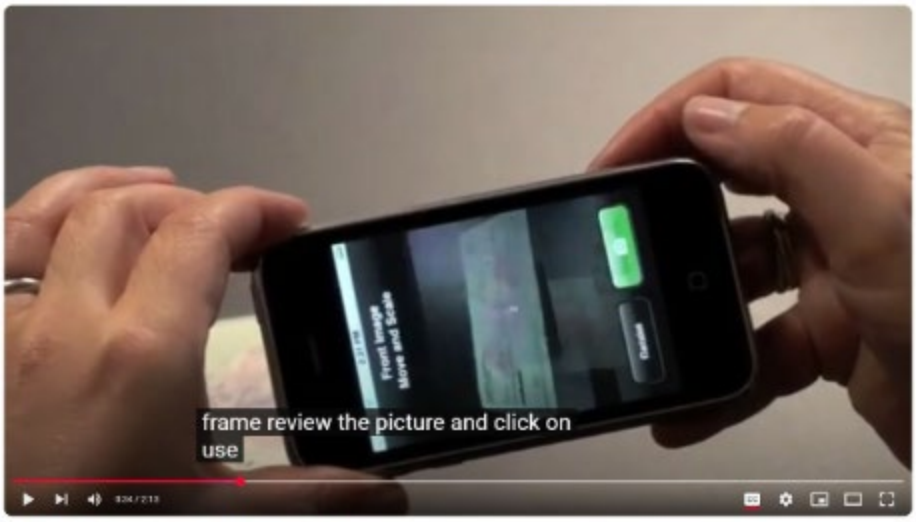
- Android Developers Website (Camera) at 1 ("The Camera class is used to connect/disconnect with the camera service, set capture settings, start/stop preview, snap a picture, and retrieve frames for encoding for video."), 2 ("setPreviewCallback(Camera.PreviewCallback cb) Can be called at any time to instruct the camera to use a callback for each preview frame in addition to displaying it. … takePicture(Camera.ShutterCallbackshutter, Camera.PictureCallback raw, Camera.PictureCallback jpeg) Registers a callback to be invoked when a picture is taken.").

- Android Website (android.hardware.CameraDevice) at 1 ("android.hardware.CameraDevice The CameraDevice class is used to open/release the camera driver, set image parameters and capture a preview frame (320*240 pixels) or a full-resolution image (1280*1024 pixels). This class does not have any public constructors. However, calling the static method open() will return a CameraDevice object. To get a frame/picture, follow these steps: open() the driver. Open will return null if it fails. Set the image parameters (using setCaptureParams(CameraDevice.CaptureParams)). Look at CameraDevice.CaptureParams class for details. Capture (capture(Canvas)) an image with a Canvas object. Note that the Bitmap in the Canvas object must already have enough space allocated for the frame/image. If you are shooting preview, then you would want to call capture(Canvas) several times. It's not necessary to set parameters after each capture(Canvas) if the parameters are the same. Finally, close() the driver.").

- Garcia at 1:6-13 ("The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.").

- CAM (Parikh 2005a) at 1 ("In this paper we introduce mobile document processing as an accessible and locally harmonious way to provide information services to the developing world [2]. In our system a camera-equipped mobile phone is used for image capture and data entry. Document interaction is specified using CamShell - a narrative document-embedded programming language that can allow developers to add interactive audio, data entry, validation, processing, and networking instructions to otherwise inert paper documents."); Parikh 2007 at 53-54 ("For barcode recognition and generation, I am using the open source toolkit developed by Rohs and Gfeller [128]. This includes a 2D barcode recognition library for Nokia Series 60 phones. A user can capture a barcode by taking a picture of it using the mobile phone [Figure 4.3] camera. Each barcode contains 83 bits of data, 7 of which are being used for error correction (leaving 76 bits of data capacity). Several barcodes can be detected from a single camera image.").

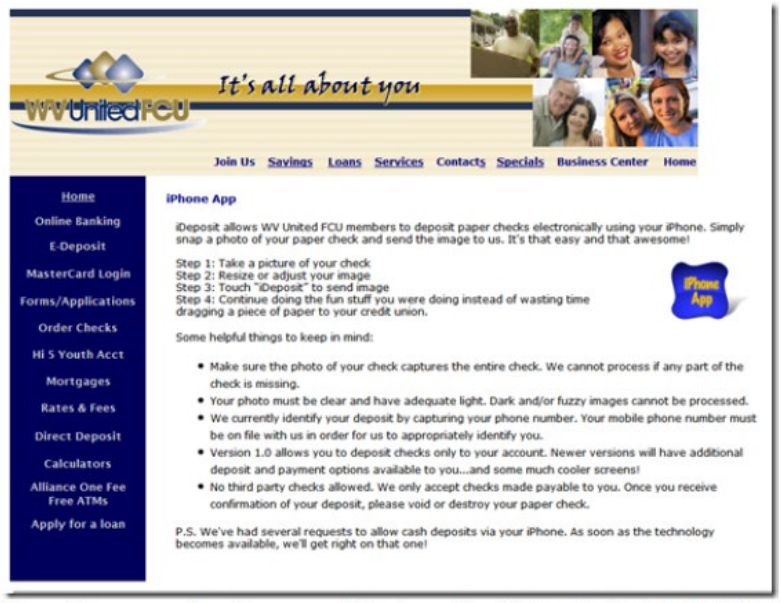
- ImageNet, Mitek - Mobile Online Banking - Depositing your check explained at 0:22-0:55.




Mobile Online Banking: Depositing your check *Explained* I Mitek

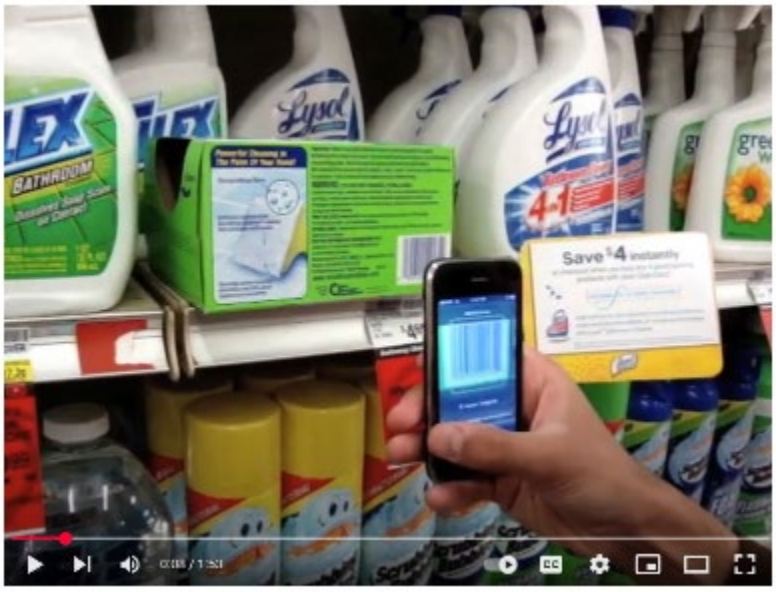
- WVUFCU iPhoneApp at 1; Finovate Blog at 4:

**iPhone app landing page (link (http://www.wvunited.org/iphone.htm))**

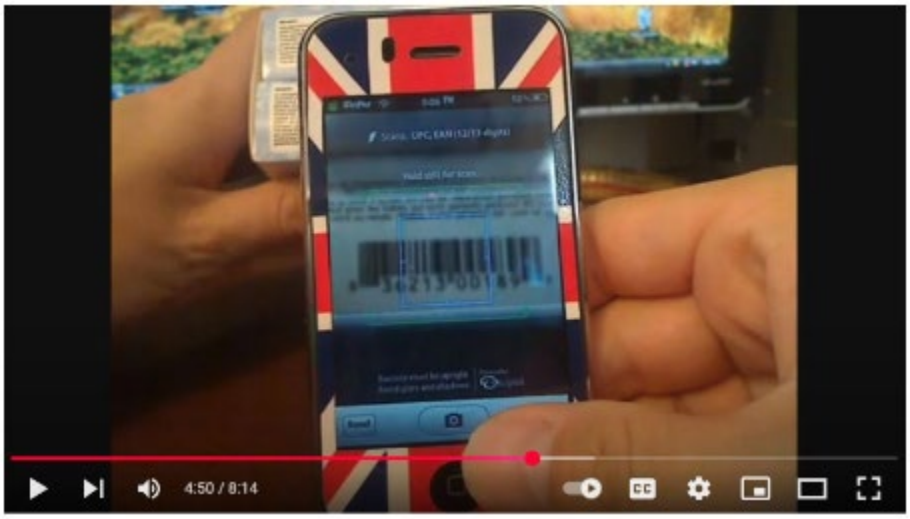



(http://www.netbanker.com/WindowsLiveWriter/a40dc3cbb662_7C06/image_8.png)

- RedLaser (Realtime Video) at 0:01-0:09




- RedLaser (Review Video) at 3:53-5:19




- ScanZoom (Attia) at 3:45-52 (“The user then takes a picture of the barcode using the digital camera. As soon as the barcode is captured, the ScanZoom software decodes the barcode utilizing a decoding engine integral to the ScanZoom software. Alternatively, the Scanzoom software may send the image of the barcode to a central server for decoding by sending a multimedia message service (“MMS”) message to the server containing the barcode image.”), Interview Video at 0:00:55



### i. Alignment Guides (’310, ’095 Patent)

Before the Asserted Patents, it was well known to use display an alignment guide on a screen to help a user take a higher quality picture, including using mobile device cameras for capturing images of documents. For example:

- Luo at Abstract (“A method and system for capturing the frontal image of an object having at least two substantially straight edges (205), which helps reduce projective distortion in the image. The method includes receiving a preview image on an image sensor (115) related to a camera (100), and the preview image depicts a straight edge (205). Then, a preview image depicting a straight edge (205) and at least two reference lines (135) simultaneously present is displayed in a preview window (125) of the camera system (100), where each reference line (135) corresponds to a straight edge (205). The image of the object is captured only when the straight edge (205) shown in the preview window (125) is substantially parallel to the corresponding reference line (135).”), FIG. 2 (“

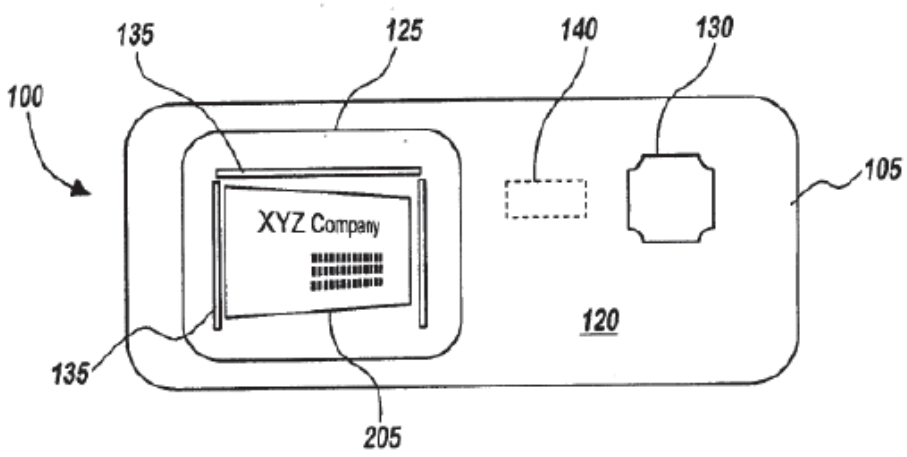



Fig. 2

").

- Yoon at [0019] ("FIG. 1 is a block diagram of a portable terminal according to the present invention. Referring to FIG. 1, the portable terminal includes a controller 100, a camera unit 102, an image processor 104, and a display unit 106. The controller 100 performs processing or controlling in order to establish a voice/data communications. In particular, in order to photograph a business card, the controller 100 receives brightness data from the camera unit 102, checks a brightness environment under which the business card is to be photographed, and controls whether to perform automatic photographing on the business card. Also, the controller 100 displays on the display unit 106 reference boundary lines, for presenting a size and location, of a business card, which are appropriate for recognizing the business card; and determines whether the boundary lines of the business card received from the image processor 104 coincide with the reference boundary lines. If they coincide with one another, the controller 100 controls automatic photographing to be performed."), FIGS. 3A-3D.

- Zhu at Abstract ("A method of scanning business cards using a mobile phone with a digital camera; the method, which pertains to the technical field of mobile phone apps, uses said mobile phone to add useful information included on a business card, such as name and address, directly to the address book of a mobile phone and includes steps for photographing the business card using the mobile phone: detecting the four sides of the targeted business card on the mobile phone screen, using lines of one color to draw the detected four sides of the targeted business card so that the mobile phone user can clearly see the position and shape of the business card on the screen, and using arrows based on the current position of the business card to prompt the mobile phone user to move the business card in the direction pointed to by the arrows until the business card arrives at a position suitable for photographing; when the business card is in a position suitable for photographing, the four sides of said business card become a second color, prompting the mobile phone user to prepare to take a photograph; to photograph the targeted business card, the mobile phone user needs only to hold the mobile

phone steady and trigger the "shoot" button, either manually or automatically by the mobile phone, thus overcoming a drawback, namely the tendency of the camera to shake while taking a photograph."), FIG. 3.

- Lin at [0153] ("The payment information also may be acquired from a credit card using a camera instead of near field communication. FIG. 13 illustrates a series of screens for acquiring payment information using the camera 46 shown in FIG. 2. From the card entry screen 218, the merchant may select the camera graphical element 236 to acquire the payment information using the camera. A camera input screen 300 facilitates entry of the payment information using the camera. The screen 300 includes a frame 302 consisting of two corners disposed on the screen 300 to identify the borders of the received image. The frame 302 may function as a viewfinder for the camera. In certain embodiments, the frame 302 may be omitted, and the entire display area may be used to receive the image.").

- CAM (Parikh 2007) at 53-54 ("For barcode recognition and generation, I am using the open source toolkit developed by Rohs and Gfeller [128]. This includes a 2D barcode recognition library for Nokia Series 60 phones. A user can capture a barcode by taking a picture of it using the mobile phone [Figure 4.3] camera. Each barcode contains 83 bits of data, 7 of which are being used for error correction (leaving 76 bits of data capacity). Several barcodes can be detected from a single camera image.").

- CAM (Parikh 2007) at 54 (Figure 4.3)

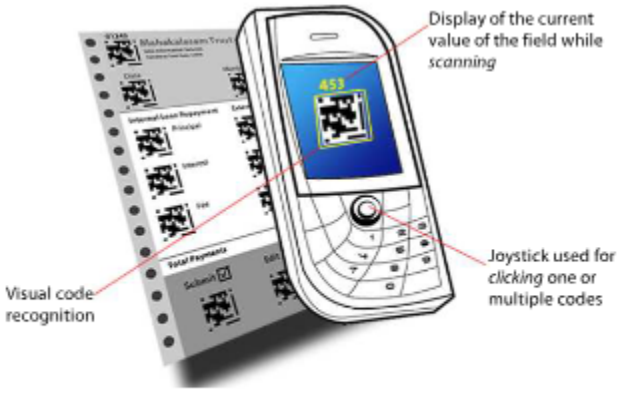
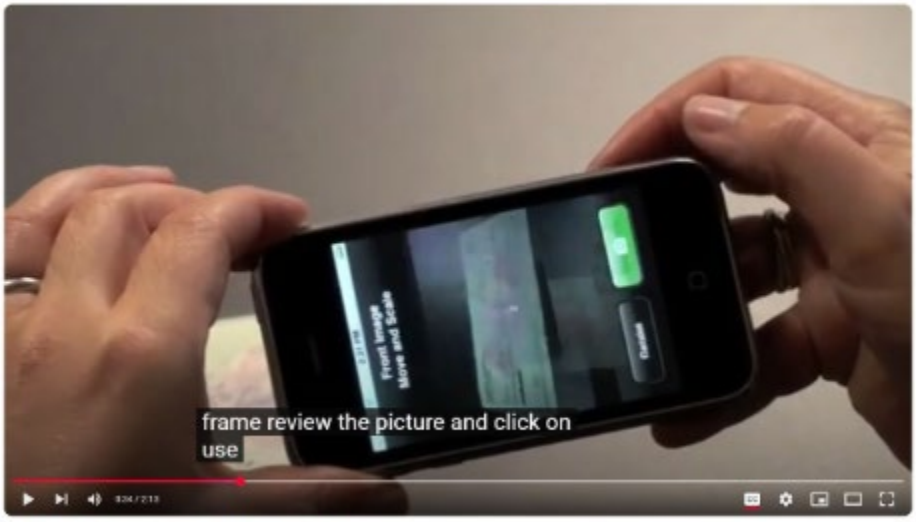



Figure 4.3: The CamBrowser application. The current value for the field is displayed beside the visual code.

- CAM (Video) at 1:44-1:53

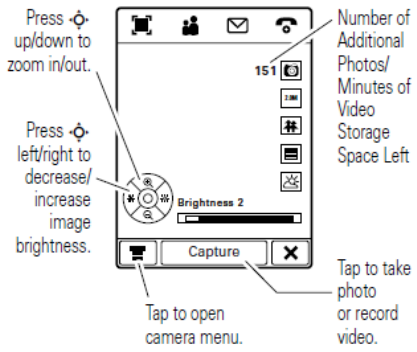



- ImageNet, Mitek - Mobile Online Banking - Depositing your check explained at 0:22-0:55.




- MotoRokr (MotoRokr Manual) at 108:

Press the camera key (see illustration on page 2) to activate the camera. The active viewfinder image appears in the display.




## use the viewfinder

Press ·Ö· or the side up/down keys to zoom in/out. Press ·Ö· left/right to adjust image brightness.

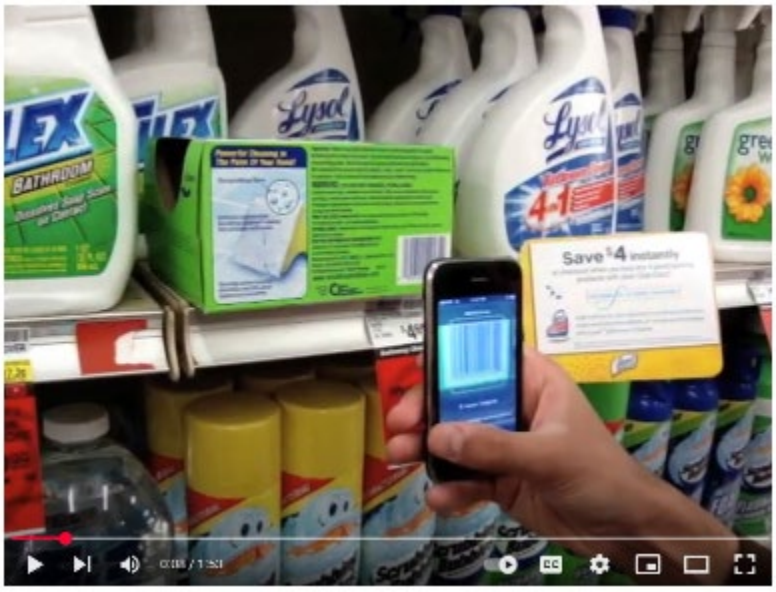
**Tip:** Tap the viewfinder screen to remove the indicators for an unobstructed fullscreen view. Tap the fullscreen view to return the indicators.

The viewfinder indicators are active and can be used to change the most-used functions (such as image size, camera/video mode, and effects mode).

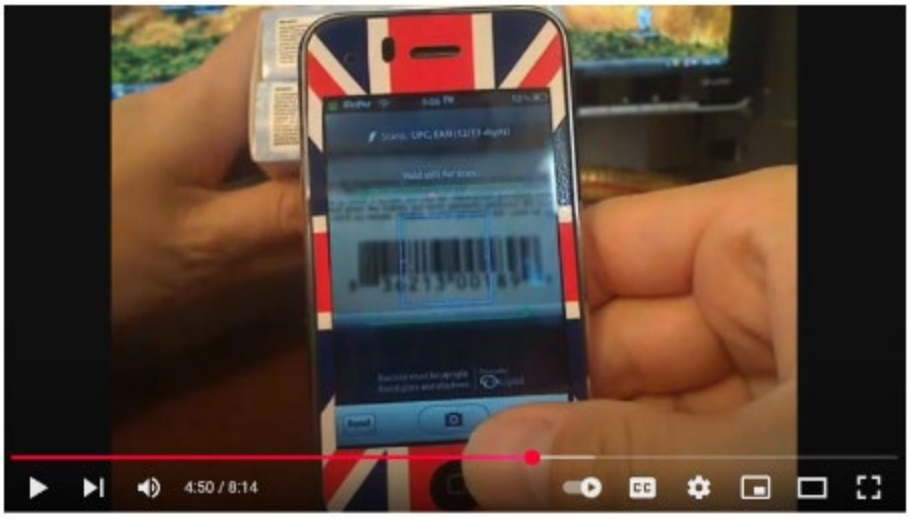
The number next to the photo/video mode indicator represents the number of additional photos, or the minutes and seconds of additional video, that can be stored in the current selected memory storage location.



- RedLaser (Realtime Video) at 0:01-0:09




- RedLaser (Review Video) at 3:53-5:19




- ScanZoom (Attia) at 6:51-7:6 ("Referring next to FIG. 2A, shown is the process utilized by ScanZoom to acquire and decode barcode 103 once the application has been launched. A user first launches the ScanZoom program on mobile device 105 by selecting its icon from a list of programs located on the device in step 201. Generally, this will initialize the camera and all the network resources needed by ScanZoom for the application to function properly. Once the user has started the program, a message appears directing the user to take a picture of barcode 103 in step 203. ScanZoom provides the user with a "preview" window which functions as a viewfinder, allowing the user to properly center and align the barcode before taking a picture. Once the barcode image has been acquired by digital camera 107, ScanZoom attempts to decode the barcode utilizing its proprietary barcode decoding algorithm in step 205. If the software cannot decode barcode 103 on a first attempt in step 205, the ScanZoom software attempts to decode the image a finite amount of times. The number of decoding attempts can either be fixed or able to be altered by the user. Once barcode 103 has been properly decoded as verified in step 207, mobile device 105 prepares the decoded information for transfer to server 113 to undergo further processing in step 209.").

**j. Automatically Capturing an Image Based on One or More Monitoring Criteria ('310, '095 Patent)**

Before the Asserted Patents, it was well known to automatically capture an image of a document when one or more monitoring criteria for the image are satisfied. For example:

- Luo at 6 ("Various techniques may be used to indicate to the user that an appropriate number of reference lines 135 are substantially parallel to the respective straight edges 205 of the object depicted in the preview window 125. For example, an alarm composed of sounds, such as a clicking sound output from the camera system 100, may be provided. In the same way, it may be the light illuminated in the preview window 125 or the light illuminated elsewhere in the

camera system 100. In addition, an image indicator may be displayed on the preview window 125, for example, when each reference line 135 is substantially parallel to a corresponding straight edge 205, the reference line 135 may change color or flicker individually. As an alternative, when the straight edge 205 displayed on the preview window 125 is substantially parallel to the corresponding reference line 135, the system 100 can automatically capture the selected image and provide the user with or without instructions.").

- Yoon at Abstract ("Provided are an apparatus and method for photographing a business card in a portable terminal. The method includes recognizing boundary lines of the business card from an image received from a camera when a recognition and photographing key for a business card is selected, and allowing the business card to be automatically photographed by using the recognized boundary lines."); [0019] ("FIG. 1 is a block diagram of a portable terminal according to the present invention. Referring to FIG. 1, the portable terminal includes a controller 100, a camera unit 102, an image processor 104, and a display unit 106. The controller 100 performs processing or controlling in order to establish a voice/data communications. In particular, in order to photograph a business card, the controller 100 receives brightness data from the camera unit 102, checks a brightness environment under which the business card is to be photographed, and controls whether to perform automatic photographing on the business card. Also, the controller 100 displays on the display unit 106 reference boundary lines, for presenting a size and location, of a business card, which are appropriate for recognizing the business card; and determines whether the boundary lines of the business card received from the image processor 104 coincide with the reference boundary lines. If they coincide with one another, the controller 100 controls automatic photographing to be performed.").

- Zhu at 6 ("Many models of mobile phones with high-resolution digital cameras currently on the market have a business card scanning feature, e.g., Motorola's "Ming" phone. This mobile phone uses another method to photograph business cards: three red lines are drawn at specific locations on the mobile phone screen to remind the mobile phone user when photographing a business card that three sides of the business card should coincide with the three red lines. When the three sides of the business card roughly coincide with said three red lines, said red lines become green lines, and the digital camera of the mobile phone simultaneously enters a countdown phase; when the countdown phase ends, the mobile phone digital camera automatically photographs the business card. The drawback to this technology is that, when a business card is photographed using a mobile phone digital camera, relative motion exists between the business card and the digital camera, and the image on the mobile phone screen continuously changes; the mobile phone user not only finds it difficult to have a clear view of the current location of the business card, but also does not know how to move the business card so as to make the business card coincide with said three red lines."), 7 ("To achieve the object described above, the present invention employs the technical solution below. While retaining steps (2), (3), and (4) of the background art above, it improves step (1) by: detecting the four sides of the targeted business

card on the mobile phone screen, drawing the detected four sides of the targeted business card using lines of a first color so that the mobile phone user can clearly see the position and shape of the business card on the screen, and using arrows based on the current position of the business card to prompt the mobile phone user to move the targeted business card in the direction pointed to by the arrows until the business card arrives at a position suitable for photographing; when the relative positions of the mobile phone digital camera and the business card are suitable for photographing, the four sides of said business card become a second color, prompting the mobile phone user to prepare to take a photograph; to photograph the targeted business card, the mobile phone user needs only to hold the mobile phone steady and trigger the "shoot" button of the digital camera, either manually or automatically by the mobile phone, thus overcoming a drawback, namely the tendency of the camera to shake during shooting."); Fig. 3.

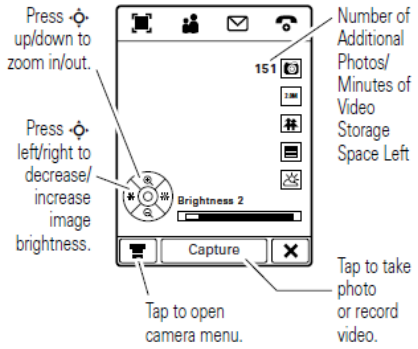
- Nepomniachtchi at Abstract ("The present invention relates to automated document processing and more particularly, to methods and systems for document image capture and processing using mobile devices. In accordance with various embodiments, methods and systems for document image capture on a mobile communication device are provided such that the image is optimized and enhanced for data extraction from the document as depicted. These methods and systems may comprise capturing an image of a document using a mobile communication device; submitting the image to a server; and processing the image to create a bi-tonal image of the document for data extraction. Additionally, these methods and systems may comprise capturing a first image of a document using the mobile communication device; automatically detecting the document within the image; geometrically correcting the image; binarizing the image; correcting the orientation of the image; correcting the size of the image; and outputting the resulting image of the document."); 11:17–27 ("Following the automatic document detection, method 1000 performs geometrical corrections to the mobile image at operation 1004. As previously noted, this can comprise cleaning up the image by performing auto-rotate operations, de-skew operations, perspective distortion correction operations, and cropping operations. Generally, this is due to perspective distortions present in the original mobile image, as well as the possibility of incorrect orientation of the document within the mobile image. The discussion of FIG. 15a will provide further detail with respect to the geometrical correction operation.").

- CAM (Parikh 2005a) at 1 ("2.1. CAMBrowser
  The CAM client application is called the CAMBrowser, and has currently been implemented for Series 60 camera-equipped mobile phones. CAMBrowser is designed to process specially designed CAMForm documents. CAMForms contain visual codes - two-dimensional data glyphs containing up to 76-bits of data that can be decoded from a camera image [3]. Visual codes serve as references to interactive content embedded within CAMForms.
  CAM interaction consists of two primitives. The user can either scan codes from low-resolution images taken by the viewfinder in real time, or click codes by taking a high-resolution image using the joystick button. Our focus is on

designing a simple and intuitive interaction model, with well-defined affordances between actions."); Parikh 2007 at 65 ("The client listens to the audio prompts to follow in the process. If she wants to verify that the data has been entered correctly, she can ask the agent to focus the camera on each of the fields to display the entered values. Once they are both satisfied, the agent captures the 'Submit' barcode to generate a MMS message. The message contains a XML file with the entered data, and an audio recording of the loan purpose. This MMS file is cached in the outgoing message queue. It will be sent when the field agent travels on the highway (a connected area) on her way to the next village. The client keeps the paper form for her own records."); Pervasive Computing at 76 ("CamBrowser The CamBrowser application's primary purpose is to capture a visual image of the form and decode and interpret the embedded CamShell program, which executes the user interface that collects information from the user. CamBrowser executes instructions in sequence, starting with instruction zero. Internally, a simple virtual machine keeps track of the current program counter and local variables. You can scan codes either by pointing the camera at the CamForm (scan mode) or by explicitly taking a picture (click mode). We've found the former to be more suitable for quick, directed user interaction, while the latter is more effective for capturing an entire CamShell program and entering large amounts of data—for example, in a loop.").

- MotoRokr (MotoRokr Manual) at 108:

Press the camera key (see illustration on page 2) to activate the camera. The active viewfinder image appears in the display.




## use the viewfinder

Press up/down or the side up/down keys to zoom in/out. Press left/right to adjust image brightness.

**Tip:** Tap the viewfinder screen to remove the indicators for an unobstructed fullscreen view. Tap the fullscreen view to return the indicators.

The viewfinder indicators are active and can be used to change the most-used functions (such as image size, camera/video mode, and effects mode).

The number next to the photo/video mode indicator represents the number of additional photos, or the minutes and seconds of additional video, that can be stored in the current selected memory storage location.



98 ("Your phone automatically captures a business card image when the card is properly framed in the OCR viewfinder. (To activate or deactivate the auto-capture feature, see "business card reader settings" on page 97).").

- RedLaser (TechCrunch) at 1

> Now that the new iPhone 3G S has a video camera, TechStars startup Occipital has released an update to its RedLaser app (iTunes link, $1.99) which speeds up barcode scanning by not requiring you to first take a picture. Occipital claims that its "realtime barcode scanner" is the only one which works on phones with both autofocus (the new 3G S) and without (the older iPhone and iPod Touch). Other barcode scanning apps, such as ShopSavvy's, can also take advantage of the autofocus camera on the 3G S, but can't do on-the-fly scanning on the older models.
>
> Like other barcode scanning apps, many of which are free, RedLaser looks up price comparisons in an ever-growing product database. The utility of these apps is really determined by how good the back-end database is more than the scanning technology itself. But RedLaser is really more of a technology demo for Occipital, which is developing some some impressive mobile visual search technologies. The scanning technology is available to other developers through an SDK.

- ScanZoom (Attia) at 7:62-8:3 ("However, if the windows message loop is empty, the user is then directed to push a button on mobile device 105 to take an image of barcode 103 in step 363. Alternatively, the image may be acquired automatically by mobile device 105 for "on-the-fly" decoding. Camera control module 305 interacts directly with API 301 to perform this step since the user must physically push a button on mobile device 105 to take the image. Alternatively, this feature may be fully automated.").

**k. Providing Instructions and/or Feedback to Users During Image Capture ('310, '095 Patent)**

Before the Asserted Patents, it was well known to provide instructions and/or feedback to a user before, during, and after capturing an image with a mobile device. For example:

- Luo at 6 ("Various techniques may be used to indicate to the user that an appropriate number of reference lines 135 are substantially parallel to the respective straight edges 205 of the object depicted in the preview window 125. For example, an alarm composed of sounds, such as a clicking sound output from the camera system 100, may be provided. In the same way, it may be the light illuminated in the preview window 125 or the light illuminated elsewhere in the camera system 100. In addition, an image indicator may be displayed on the preview window 125, for example, when each reference line 135 is substantially parallel to a corresponding straight edge 205, the reference line 135 may change color or flicker individually. As an alternative, when the straight edge 205 displayed on the preview window 125 is substantially parallel to the

corresponding reference line 135, the system 100 can automatically capture the selected image and provide the user with or without instructions.").

- Yoon at [0022] ("The display unit 106 displays status information and characters (which consist of limited numbers) generated during the operation of the portable terminal. In particular the display unit 106 outputs an image received from the image processor 104, displays the reference boundary lines for the business card under control of the controller 100, and changes and displays the color, shape or thickness of each of the reference boundary lines."); [0026] ("If it is determined that the current brightness is not appropriate for photographing the business card, the portable terminal displays a message indicating this fact on the display unit 106 in step 227, and then returns to operation 205. If it is determined that the current brightness is appropriate for photographing the business card, the portable terminal recognizes the boundary lines of the business card by analyzing an image received from the camera sensor in step 209. Next, the portable terminal proceeds to operation 211, and determines whether at least one of the reference boundary lines (upper, lower, left, and right boundary lines) coincides with the boundary lines of the photographed business card. Here, a predetermined margin of error is allowed when determining whether the reference boundary lines coincide with the boundary lines of the business card. For example, if the predetermined margin of error is 3 mm, the left boundary line of the business card can be determined to coincide with the left reference boundary line when the top (or bottom) of the left boundary line of the business card is 2 mm away from the top (or bottom) of the left reference boundary line in a vertical or horizontal direction."); [0028] ("Next, the portable terminal determines whether all the upper, lower, left and right reference boundary lines coincide with the boundary lines of the business card in step 215. If all the reference boundary lines do not coincide with all the boundary lines of the business card, the portable terminal returns to step 209. If all the reference boundary lines completely coincide with the boundary lines of the business card as illustrated in FIG. 3D, the portable terminal proceeds to step 217 and displays a message informing the user of the start of automatic photographing for the business card. Here, it is possible to inform the user of the start of automatic photographing by generating a predetermined alarm sound of automatic photographing or changing the shapes, thicknesses or colors of the reference boundary lines.").

- Acharya at 3:1–20 ("In one embodiment of the invention, the RCT is located at home, and comprises a digital camera, or a personal computer or wireless handheld device, such as a Blackberry® or Palm Treo®, having digital imaging capabilities such as an attached image scanner or a digital camera. To initiate the transaction, the banking customer uses an Internet-based application loaded on the RCT to access the BOFD system. A pre-assigned Personal Identification Number (PIN) or other password or secret, including a biometric, may be required to allow the banking customer access to the BOFD system. Either when prompted or prior to accessing the BOFD system, the banking customer captures the digital image of the financial instrument by the scanner or the digital camera and prepares a file storing the digital image. Upon prompting by the BOFD system, the banking

customer forwards the digital image to the BOFD system along with the data that was either 'recognized' from the digital image of the financial instrument or was input into the RCT by the banking customer. Alternatively, the customer may input data regarding the financial instrument directly into the BOFD system."); 7:5–13 ("In step 230, the banking customer may respond to prompts for each item of data needed to deposit the financial instrument. In one embodiment, the banking customer may enter a predetermined set of data all at once, for example where all data input fields are visible on the display screen of the RCT 100. The banking customer may be prompted to supply additional information in step 240 if the BOFD system 110 requests information not entered previously by the banking customer."); 8:16–34 ("In one embodiment, the banking customer may log onto BOFD system 110 and check the status of a deposit transaction. In another embodiment, the BOFD system 110 may notify the banking customer that there is missing or erroneous information or data regarding the financial instrument or deposit transaction request, or there are outstanding questions regarding the deposit transaction request, in which case the banking customer may log onto the BOFD system 110 and input missing or erroneous information or data or provide answers to outstanding questions. In another embodiment, the banking customer may access BOFD system 110 and access previously inputted data to check for correctness, to correct erroneous inputted data or to make other alterations to the deposit transaction request. For example, the banking customer may request that the deposit be placed in a different account than was originally requested, that some of the deposit amount be credited to an outstanding account such as a loan or a credit account or that some of the deposit amount be placed in a stored value account.").

- Zhu at 9 ("The technical solution of the present invention is further characterized in that said camera system displays on said screen instructional text guiding the mobile phone user to perform correct operations during the execution of each step."); 10 (similar); 7 ("To achieve the object described above, the present invention employs the technical solution below. While retaining steps (2), (3), and (4) of the background art above, it improves step (1) by: detecting the four sides of the targeted business card on the mobile phone screen, drawing the detected four sides of the targeted business card using lines of a first color so that the mobile phone user can clearly see the position and shape of the business card on the screen, and using arrows based on the current position of the business card to prompt the mobile phone user to move the targeted business card in the direction pointed to by the arrows until the business card arrives at a position suitable for photographing; when the relative positions of the mobile phone digital camera and the business card are suitable for photographing, the four sides of said business card become a second color, prompting the mobile phone user to prepare to take a photograph; to photograph the targeted business card, the mobile phone user needs only to hold the mobile phone steady and trigger the "shoot" button of the digital camera, either manually or automatically by the mobile phone, thus overcoming a drawback, namely the tendency of the camera to shake during shooting."); 2 ("first step: aiming the lens of the built-in digital camera of said mobile phone at the targeted business card, said camera system displaying an

image of the targeted business card on the screen of said mobile phone and separately displaying four corner points in places near the four corners on said screen, indicating to the mobile phone user that the vertices of the four corners of the targeted business card should coincide, respectively, with said four corner points when the targeted business card is in a position suitable for photographing … third step: using lines of the first color to draw the four sides of the targeted business card detected in the second step on said screen, and using indicating arrows of the first color to indicate the direction in which the targeted business card should move; fourth step: said camera system checking whether the targeted business card is in a suitable position for photographing; if not, returning to and executing the second step; if so, using lines in the second color to draw the four sides of the targeted business card on said screen to indicate to the mobile phone user that the targeted business card is in a suitable position for photographing and that it is now possible to take the photograph; proceeding to and executing the fifth step"); claims 3, 5, 7, 8, FIG. 3.

- Lin at [0153] ("The payment information also may be acquired from a credit card using a camera instead of near field communication. FIG. 13 illustrates a series of screens for acquiring payment information using the camera 46 shown in FIG. 2. From the card entry screen 218, the merchant may select the camera graphical element 236 to acquire the payment information using the camera. A camera input screen 300 facilitates entry of the payment information using the camera. The screen 300 includes a frame 302 consisting of two corners disposed on the screen 300 to identify the borders of the received image. The frame 302 may function as a viewfinder for the camera. In certain embodiments, the frame 302 may be omitted, and the entire display area may be used to receive the image."); [1054] ("Instructions 304 prompt the merchant to take a picture of the credit card. The merchant may position the electronic device 10 so that the card image is aligned within the frame 302, and then the merchant may select graphical element 306 to capture the image. If the merchant would like to recapture the image, the merchant may select the graphical element 260 to return to the previous screen 218. If the merchant has successfully captured the image, the merchant may select the graphical element 228 to begin the payment process using the captured image. Selection of the graphical element 228 may cause a CVV entry screen 308 to be displayed. Similar to the process described with respect to FIG. 11, the merchant may enter the CVV using the keypad 266. After entry, the merchant may select the pay button 274 to process the payment. The payment status screen 240 may then be displayed and an approval message may be displayed within the window 242 to notify the merchant when the transaction has been approved."), [0160] ("The check information also may be acquired through the camera. For example, the graphical element 236 may be selected to enter the check information using techniques similar to those described with respect to FIGS. 13 and 14. In certain embodiments, the check information may be acquired using the scanner 48 (FIG. 2). For example, the graphical element 336 may be selected to scan a bar code on the check using the scan device 48 (FIG. 2). In other embodiments, the payment information may be acquired by scanning an NFC enabled check. In these embodiments, a graphical element may be included within the summary bar 152

to enter the check information using near field communication. The check information may be acquired by near field communication using techniques similar to those described above with respect to FIGS. 11-12.").

- Fellerman at [0045] ("For the remote terminal 26 to perform the aforementioned capture, aggregation, and data transmission operations, it is contemplated that the remote terminal 26 is programmed with executable instructions therefor that coordinates the operation of the check scanner 28, the keypad 54, the screen 56, and the communications module 58. Further details of such executable instructions, and in particular those relating to the capture and transmission of the checks 24, are illustrated in the flowchart of FIG. 5. The method begins with a step 500 of receiving an input from the keypad 54 to initiate a deferred payment transaction. Thereafter, in step 502, an identification card of the purchaser 10 such as a driver's license or other government-issued credential with a magnetic encoding strip is scanned with a magnetic card reader 62. Typical information encoded in a magnetic strip of an identification card includes an identification number, date of birth, and the issuing State, among others. In the absence of a magnetic encoding strip, it is also contemplated that a barcode, if present on the identification card, may be scanned with a bar code reader to capture the same information. If it is determined in the decision block 504 that the read was not successful or that the presented identification card did not have a magnetic strip, the method continues with manual entry steps 506, 507, and 508, where the keypad 54 is used to enter the identification number, a code corresponding to the issuing State, and the date of birth, respectively. Otherwise, such data is read directly from the identification card. Preliminary data collection concludes with a step 510 of entering the phone number of the purchaser 10 via the keypad 54. Throughout each of these data collection steps, the screen 56 displays corresponding prompts therefor to request proper data from the user/merchant 14."), [0046] ("Next, in step 512 the check number 44 from the first of the plurality of checks to be scanned is entered via the keypad 54, and in step 512, the amount specified thereby is likewise entered via the keypad 54. Again, the screen 54 displays prompts corresponding to the respective data inputs. Then, the check scanner 28 is activated and the check 24 is fed for scanning in step 516. After scanning the front and back sides of the check 24, basic verifications are performed according to decision block 518 to ensure that the image data was properly captured. For example, if the check scanner 28 timed out before receiving any data, then an indicator would be activated to prompt a re-scan. Although the check number and check amount data entered into the remote terminal 26 via the keypad 54 corresponds to the scanned check images, the check images are understood to be stored independently in certain embodiments."), [0048] ("All of the entered data is then verified in a check summary confirmation step 524 and a detail confirmation step 526. In one embodiment, the total of all the checks 24 presented as keyed into the remote terminal 26 in step 514 may confirmed. However, other related aggregate or summary data relating to the checks 24 may also be confirmed. Those having ordinary skill in the art will readily devise such alternative summary data. In the detail confirmation step 526, all or part of the entered data, such as the identification number, date of birth,

issuing state, phone number, the check number for each of the scanned checks, and so forth may be verified.").

- Viera at 7:33-64 ("FIGS. 5A-E are example graphical user interfaces illustrating a deposit transaction according to another embodiment of the present invention. FIG. 5A illustrates a menu of services 501 that is presented to the user. The user will select a service by touching a labeled button 502. If the user wants to end the transaction, he or she can touch the exit button 503. FIG. 5B illustrates a listing of the user's accounts 504. Each account is displayed with the account type 505, the balance 506, and the available funds 507. The user selects the account that will be receiving the deposit. FIG. 5C illustrates a check deposit 508. A check is inserted into an image scanner without an envelope. The account type 509 is displayed. Verification of check scan 510 is displayed. The user uses a keypad 511 to enter the deposit amount 512. An image 513 of the check is displayed to the user. FIG. 5D illustrates an appraisal of funds availability. The account type 515 is displayed. The deposit amount 516 is also displayed. The funds that are immediately available are displayed at 517A. The availability of the remaining funds is displayed at 517B. The user can reject the deposit by touching button 519 or button 518 to accept. If the deposit is accepted, the deposit information is stored in a remote database that may be accessed by the user's financial institution. For example, after the transaction has completed, FIG. 5E illustrates instant online review of the deposit. A financial institution employee can review the details of the deposit and also see an image of the actual check. Previously, a check must be placed into an envelope for deposit. The envelope then sits in a depository until a teller can process the deposit and determine the availability. A user can then find out what the funds availability is through contacting the financial institution. The institution may also send a letter to the account holder's home with the funds availability.").

- Cohen at [0072] ("The scan process entails placing one or more checks or check/stub units of work into the attached scanner. The initiation of the scanning process begins to feed strictly checks, or a combination of checks and payment coupons, through the transport, where images and MICR information are captured and displayed on the computer screen for verification and possible removal and rescanning, as previously described in detail hereinabove. It is at this time that any physical endorsement of the item occurs. Scanning of the associated coupon(s) and recognition of the account identification allows the payments to be posted to the Payor(s) account(s) absent, or with limited, human intervention. This process utilizes proprietary code developed by the present applicant. It also uses scanner interface software, licensed from Silver Bullet, Inc., which allows the scanner software programs to interface with a variety of scanning transports, ranging from small desktop scanners to multi-pocket transports capable of processing hundreds of items per minute."), [0078] ("Based upon the confidence levels for each item, and values defined within the profile, a Correct and Verify program module displays various items on screen and allows the user to view and override values provided by the Recognition program module.").

- Byrne at [0058] (“The images are also sent, for example, via an wfXML Service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending Such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition.”).

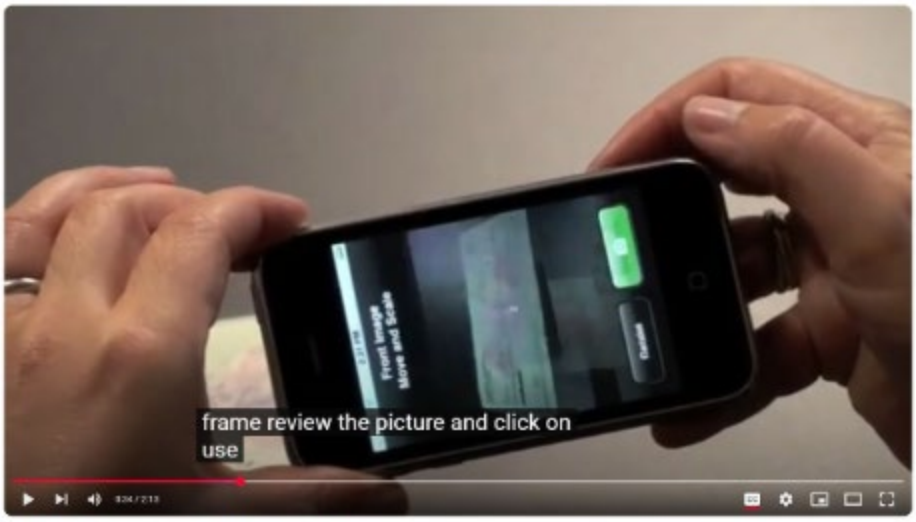
- Garcia at 6:6-7 (“The immediate displaying of the digital image by the user.”).

- Nepomniachtchi at 9:22–29 (“In operation 702, in the illustrated embodiment, the type of document is selected. For example, a user might select a document type for a check, payment coupon or deposit slip. By entering the type of document, a mobile device might be able to scan specific parts of an image to determine, for example, payee, check amount, signature, etc. In some embodiments, however, a device might determine what type of document is being imaged by processing the image.”); 9:30–46 (“In operation 704, an image is captured using, for example, a mobile communication device. In the illustrated embodiment an application running on the mobile communication device may prompt the user of the device to take a picture of the front of the document. The back of the document might also be imaged. For example, if the document is a check, an image of the back of the document might be necessary because the back of the check might need to be endorsed. If the back of the document needs to be imaged, the application may prompt the user to take the image. The application might also conduct some image processing to determine if the quality of the image or images is sufficient for further processing in accordance with the systems and methods described herein. The quality needed for further processing might vary from implementation to implementation. For example, some systems might be better able to determine information contained on a poor quality image then other systems.”); 10:18–28 (“In operation 712, the processing of the document using a mobile device is completed. For example, when the server has confirmed that all necessary data can be extracted from a received image, it might transmit a status message to the mobile device that transmitted the image. Alternatively, if some necessary data cannot be extracted, the server may transmit a request for additional data. This request might include a request for an additional image. In some cases, the request may be for data entered by a user, for example, an amount, e.g., of a check, that might be entered using a key pad on the mobile communication device.”); 10:29–48 (“In some embodiments, the quality of the image is determined at the mobile device. In this way the number of requests from the server for additional images might be reduced. The request may come directly from the mobile device. This

may allow for the request to be more quickly determined and may allow a user to take an additional image within a shorter time from the earlier image. This may mean, for example, that the user is still physically close to the document and is still holding the communication device. This might make it easier to retake an image. If the image quality processing occurs at a server it might take longer to determine that the image quality is acceptable and communicate that information back to a user. This may mean the user is no longer near the document or has started performing another task. It will be understood, however, that in some embodiments, a server based implementation might be employed to off-load processing demands from the mobile device. Additionally, in some cases it might be quick as or quicker than a system that uses the mobile communication device to process an image to determine image quality.")

- CAM (Parikh 2005b) at 2 ("CamShell provides an API for accessing the mobile phone's functionality - including the phone's user interface, networking and telephony features. Listed below are the main functions currently included in this API:
  • User Dialogs - These instructions launch various user dialogs to collect data, get confirmation or convey a message to the user. Each dialog can be associated with audio and graphical prompts. Audio feedback has been found very useful by potential users.
  • HTTP Post, Get - These instructions are used to make HTTP requests using the phone's built-in connectivity.
  • SMS, MMS, Email - These instructions transmit asynchronous network messages. In the case of MMS and Email the current form image can be attached.
  • Phone Call - This instruction is used to make a phone call to a particular number.
  • Applications - These instructions are used to launch other applications, such as the Web or WAP browser."); Parikh 2007 at 58-59 ("After data has been entered, the user can review values before submitting them to the server. By focusing the mobile phone camera on the appropriate barcode, the currently entered data is displayed on the screen, usually next to the value entered on paper (see figure 4.4). This allows both the record keeper and the SHG member to confirm that data was entered properly (mimicking the transparency of the earlier paper-based process). By focusing on the loan purpose field, the recorded audio clip is played. If a value is incorrect, the user can capture an image of just that barcode, displaying a prompt to edit the value.").

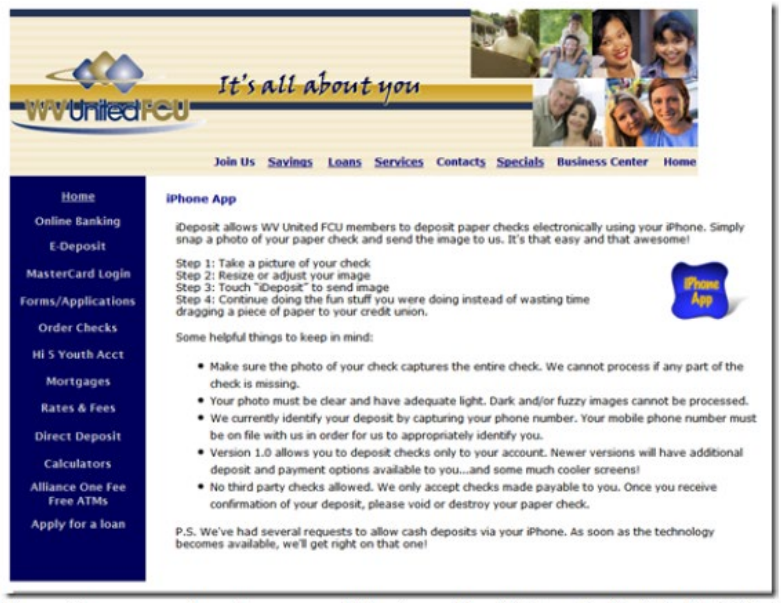
- ImageNet, Mitek - Mobile Online Banking - Depositing your check explained at 0:22-0:55.




Mobile Online Banking: Depositing your check *Explained* | Mitek

- WVUFCU (iPhoneApp) at 1; Finovate Blog at 4:

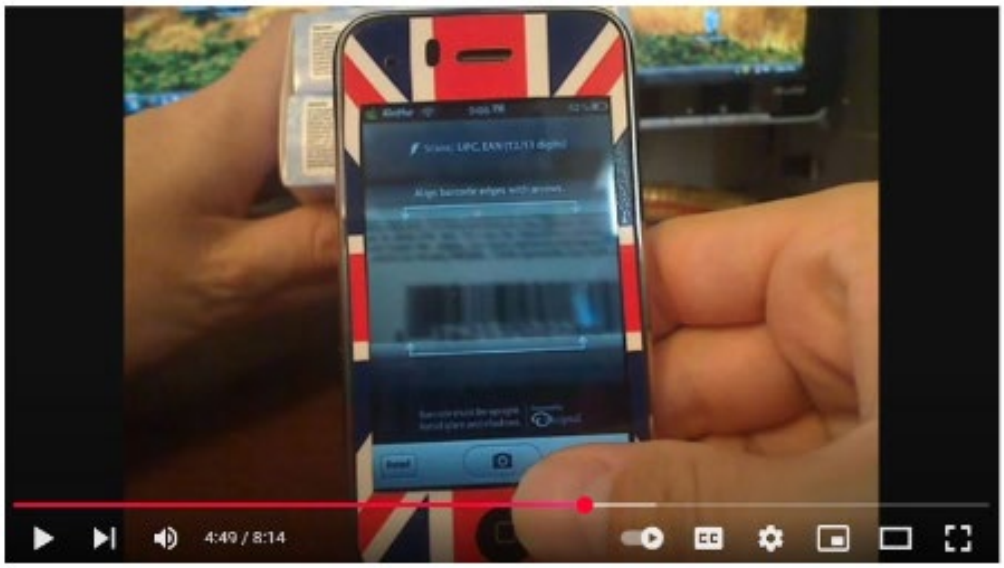
**iPhone app landing page (link (http://www.wvunited.org/iphone.htm))**




(http://www.netbanker.com/WindowsLiveWriter/a40dc3cbb662_7C06/image_8.png)

- MotoRokr (MotoRokr Manual) at 97 ("Use the red border in the OCR viewfinder to vertically frame the business card you want to capture. Position the camera so that the card fills the frame without extending beyond the frame borders. The framing border and center indicator turn green when the business card is properly framed.").

- RedLaser (Review Video) at 4:49




### l. Wirelessly Transmitting an Image of a Check ('310, '095 Patent)

Before the Asserted Patents, it was well known to transmit an image of a check to a bank for deposit, including by transmitting an image captured by a mobile device over a wireless/cellular network. For example:

- Garcia at 4:11-18 ("The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations."), 11:20-12:3 ("The communication link between each user and the institution consists of a mobile telephone communications network, for example a GSM/GPRS network or any future technology with the technological capabilities adapted to the present and future demands of the method of reference.").

- Acharya at Abstract ("A system and method for initiating a deposit transaction, where the depositor is a banking customer located at a remote location, and where the item to be deposited is a financial instrument, such as a paper check from a third party, payable to the depositor. The enabling system features a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. An image and/or other data of the financial instrument are transmitted from the RCT to the Bank of First Deposit (BOFD) where the data may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction. The deposit transaction can be accomplished without physical transfer of the financial instrument to or through the bank system. The system and method provide convenience and improved transaction

processing speed compared to other deposit transactions of financial instruments."); 4:18 ("The RCT 100 may be a telephone, digital camera, fax machine, personal computer, ATM, cell phone, PDA or any other computer, apparatus, wireless handheld device such as a Blackberry® or Palm Treo® or system capable of collecting data and communicating with BOFD system 110. For instance, the RCT 100 may comprise a workstation or other device running the Microsoft Windows™ NT™, Windows™ 2000, Windows Vista™, Windows XP™, Windows Mobile™, Unix, Linux, Xenix, IBM AIX, Hewlett-Packard UX, Novell Netware™, Sun Microsystems Solaris™, OS/2™, BeOS™, Mach, Apache, OpenStep™, Mac OS X™, GAME BOY™, PXP™ or any other operating system or platform now known or later developed. RCT 100 may also comprise two or more such devices.").

- Lin at [0162] ("In certain embodiments, the approval step may include transmitting the check information to a verification service for approval. For example, the merchant may hire a third party check verification service to verify checks received from customers. In these embodiments, selection of the graphical element 228 may cause the device to send the check information to the external server 96 (FIG. 4), which may in turn send the check information to the check verification service. The check verification service may approve the check for payment and transmit the approval to the electronic device 10 through the external server. Upon receipt, the payment authorization may be displayed within the window 242. The check information also may be transmitted to the customer's bank to verify that the customer has sufficient funds to process the transaction. Further, in other embodiments, the approval process may simply include verifying that all the check information has been entered, and no additional verification may be performed. After the transaction has been approved, the merchant may select the POP graphical element 224 to conduct another transaction.").

- Rodriguez at [0031] ("E-check transmitter 138 may transmit the e-check to purchaser's bank 140 to request payment. In many embodiments, POS terminal 130 may transmit the e-check directly to purchaser's bank 140 for payment via LAN/WAN 160 and purchaser's bank 140 may transmit, e.g., by electronic funds transfer (EFT), the funds to merchant's account 152 via LAN/WAN 160. In further embodiments, POS terminal 130 may transmit the e-check to merchant's bank 150 and merchant's bank 150 may submit the e-check to purchaser's bank 140 for payment. For example, upon generation of an e-check for check 115, e-check transmitter 138 may associate an electronic address such as an Internet Protocol (IP) address with the e-check. In particular, e-check transmitter 138 may identify an IP address for purchaser's bank 140 based upon a payer's routing number such as routing number 212 and transmit the e-check to purchaser's bank 140 via LAN/WAN 160. "); [0048] ("One embodiment of the invention is implemented as a program product for use with a computer system such as, for example, the system 100 shown in FIG. 1. The program(s) of the program product defines functions of the embodiments (including the methods described herein) and can be contained on a variety of signal-bearing media. Illustrative signal-bearing media include, but are not limited to: (i) information permanently stored

on non-writable storage media (e.g., read-only memory devices within a computer such as CD-ROM disks readable by a CD-ROM drive); (ii) alterable information stored on writable storage media (e.g., hard-disk drive or floppy disk within a diskette drive); and (iii) information conveyed to a computer by a communications medium, such as through a computer or telephone network, including wireless communications. The latter embodiment specifically includes information downloaded from the Internet and other networks. Such signal-bearing media, when carrying computer-readable instructions that direct the functions of the present invention, represent embodiments of the present invention.").

- Fellerman at [0051] ("Upon approval, the method continues with the transmission of the check images. Referring to the flowchart of FIG. 3, one embodiment of the present invention envisions a step 220 of transmitting the images of the checks 24 to the central payment processor 52 immediately after the receipt thereof. The check scanner 28 may independently store and transmit the image data, so it is understood that the remote terminal 26 generates instructions or commands to the check scanner 28 that initiate the transmission. Because the transaction has already been approved before transmitting the checks 24 to the central payment processor, it is not necessary to do so before the purchaser 12 accepts the goods or services. Instead, the images may be transmitted in a batch accumulated after a day of business. It will be appreciated that the batch transmission may proceed on any interval, such as every other day, weekly, biweekly, and so forth."), [0044] ("In further detail, the communications module 58 may be any one of widely known data communications modalities. For example, it may be a modem device that establishes a modulated audio data communications link 60 to the central payment processor 52 over standard telephone lines. Alternatively, it may be an Ethernet interface that connects to the central payment processor 52 over the Internet or other high-speed data network. Other envisioned data communications modalities include cellular telephone links, WiFi links, and the like. According to some embodiments of the present invention, the check scanner 28 establishes an alternative, independent network connection to the central payment processor 52 for the transmission of check image data.").

- Viera at 3:1-16 ("Financial kiosk 110 may include software and hardware components, such as a user interface 113 and devices 112 controlled by session unit 111. User interface 113 may include a display and touch screen for providing input and output interface functionality. Session unit 111 may include software components for managing user transaction flow. Device 112 may include a card reader, check scanner, check printer, and optical character recognition scanner. Request manager 114, on the client side, may include software and hardware components for sending and receiving procedures to and from the request manager 120 on the server side. Request manager 120 may include software and hardware components for controlling request activities between kiosk 110 and server 130 through an Internet-based network 121. Server 130 may include software and hardware components to enable communication between kiosk 110, database 131, and host 140. Server 130 processes data received from kiosk 110. It

may send the data to be stored in or queried against database 131. It may also send data over an Internet-based network 132 to a shared branching network 150 through a host 140. Database 131 is a web-maintained central depository of policies and configuration items.").

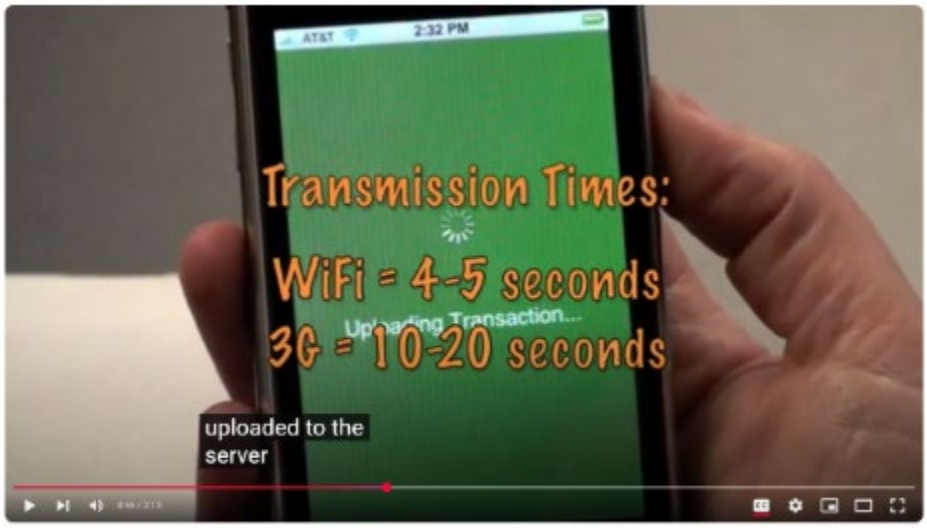
- Cohen at [0024] ("Referring now to FIGS. 1 and 4, the DirectFed™ clearing system 118 includes a Reporting System component 126, providing a reporting process 400. Initially, in step 402, reports are generated for delivery to the merchant, the merchant's BOFD 128 and/or a community bank's processor (for local archiving), via secure Internet 116 transmission. Reports preferably detail the check information, the timing and location of completed check deliveries, and processing type (i.e., Check 21 or ACH). Preferably, an Internet website operated from or by the DirectFed™ clearing system 118, enables financial institutions, merchants, and any other predetermined entity or party, to review the status of all checks uploaded to database 121, as well as any other desired related data or information.").

- Byrne at Abstract ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); [0003] ("The invention relates to providing a plug-in to drive a device for securely transmitting documents in real time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to Scan checks using a personal computer and deposit the checks via the browser to a bank account."); [0033] ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to Scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); [0041] ("A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); *see also* FIGs. 1-7.

- Nepomniachtchi at 6:8–16 ("In some embodiments, checks 100, payment coupons 200, or other documents might be imaged using a mobile device. The mobile device may be a mobile telephone handset, Personal Digital Assistant, or other mobile communication device. The mobile device may include a camera, or might include functionality that allows it to connect to a camera. This connection might

be wired or wireless. In this way the mobile device may connect to an external camera and receive images from the camera.").

- ImageNet, Mitek - Mobile Online Banking - Depositing your check explained at 0:22-0:55.



Mobile Online Banking: Depositing your check *Explained* | Mitek

- WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("Users take a picture by pressing the button on the left (see inset), verifying that the image is readable, then uploading with the button on the rightr."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

**m. Receiving an Indication of a Result of Processing of a Check Image ('310, '095 Patent)**

Before the Asserted Patents, it was well known to receive an indication of a result of processing of a check image, including receiving such indication by a user device either after processing has concluded on the user's device or at a remote location. For example:

- Garcia at 12:22-13:5 ("Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.").

- Cohen at [0022] ("Referring now to FIGS. 1 and 2, at the conclusion of either the scanning or data importation process step 202, in step 204 another process is

triggered which performs OCR as well as parsing of the MICR Line information. In a subsequent step 206, the OCR and parsing processes return value data, including: (1) check amount; (2) bank routing number; (3) bank account number; and (4) check number. Significantly, these processes also return quantitative measurements, or values, representing the confidence level of the accuracy of the OCR results pertaining to each of the respective pieces of data. Subsequently, in step 208, a remote operator at the local merchant site 100 reviews the images and data for each check in the batch. Significantly, the efficacy of this process is greatly enhanced by assigning color-coding in accordance with the respective confidence levels. Preferably, green, yellow and red colors are utilized to provide easy visual indication to the operator as to the confidence level attached to each data item. Subsequently, in step 210, and based upon a configuration setting in the profile that governs the particular batch, the operator is required to manually validate only those data items with a certain confidence level. Termed "Examination Rigor", this profile configuration value may be set to require that only low confidence data (i.e. marked red) need be inspected, or that both low and moderate confidence level data (i.e marked red or yellow) must be inspected, or that all data (i.e. even items marked green) must be inspected for errors. Upon completion of the review, the batch is ready for transmission to the DirectFed™ clearing system 118, or a like third party system. For convenience, all further description herein will refer to transmission to the DirectFed™ clearing system 118. In step 212, the image and data information is packaged at the client 100 for subsequent transmission, in step 214, to the DirectFed™ clearing system 118 via the Internet utilizing an encrypted web service."), [0053] ("Step 14: The DirectFed™ reporting system generates a report (in Skyline Data proprietary format) for delivery to the merchant. This report gives the total deposit of the merchant, subtotals by bank, subtotals by account number, subtotals by merchant location, as well as detailed information from each check deposited.").

- Acharya at Claim 20 ("[R]eceiving confirmation that the value of the financial instrument has been deposited into the account of the banking customer by the automated electronic banking system without physical receipt of the financial instrument by the automated electronic banking system.").

- Lin at [0161] ("The screen also includes graphical elements 334 and 228 for navigating between screens. Specifically, the graphical element 334 allows the merchant to return to the payment screen 198 to receive a different form of payment. Once the check information has been entered, using the keypad 266 (FIG. 11), the camera 46 (FIG. 2), the scanner 48 (FIG. 2), or the NFC device 44 (FIG. 1), the merchant may select the graphical element 228 to process the payment and advance to the payment screen 240. As noted above, the payment screen 240 includes the display window 242 indicating the status of the transaction.").

- Rodriguez at [0034] ("Once e-check clearer 142 determines that the funds are available in purchaser's account 144, e-check clearer 142 may transmit a confirmation to POS terminal 130 and transfer the funds to merchant's account

152 via LAN/WAN 160. In other embodiments, purchaser's bank 140 may transfer the funds to merchant's account 152 and merchant's bank 150 may confirm the receipt of the funds with POS terminal 130.").

- Fellerman at [0043] ("Generally, the remote terminal 26 is a data processing apparatus for capturing, aggregating, and transmitting business transaction data to a remote system for completing payment. The remote terminal 26 includes a multifunction keypad 54, a screen 56 for displaying the results of the processed transactions and associated status messages, and a communications module 58. In addition to the foregoing, the remote terminal 26 may include a magnetic card/strip reader 36, as well as a printer 63 for reasons that will be described more fully below. By way of example only and not of limitation, the remote terminal 26 is an Omni Vx terminal from VeriFone, Inc. of San Jose, Calif., though any other suitable remote terminal device may be utilized. Depending upon the specific remote terminal 26, the above-described components may be internal thereto or connected to the terminal 26 as an external peripheral device. "), [0048] ("All of the entered data is then verified in a check summary confirmation step 524 and a detail confirmation step 526. In one embodiment, the total of all the checks 24 presented as keyed into the remote terminal 26 in step 514 may confirmed. However, other related aggregate or summary data relating to the checks 24 may also be confirmed. Those having ordinary skill in the art will readily devise such alternative summary data. In the detail confirmation step 526, all or part of the entered data, such as the identification number, date of birth, issuing state, phone number, the check number for each of the scanned checks, and so forth may be verified."), [0055] ("As indicated above, the remote terminal includes an integrated printer 63 to produce the receipt 110. The printer 63 may be any one of many widely utilized types, including thermal transfer, impact, and inkjet printers. As best illustrated in FIG. 8, the receipt 110 includes all of the pertinent information previously entered, as well additional details pertaining to the merchant 14. The receipt 110 may be segregated into multiple blocks, including a merchant identifier block 112 setting forth the name and address of the merchant 14. Additionally, a time and date stamp 114 is included that details the time and date the transaction was entered. A transaction details block 116 includes a unique identification number of the remote terminal 26, the batch number of the transaction, and the routing number 40 and account number 42 of the checks 24 presented. An itemized check listing 118 includes a column for an item identification number, the amount and the check numbers 44 of the checks 24 presented, and the approval codes therefor. A total dollar amount for the entire transaction is also included. A notice block 120 highlights the most pertinent terms from the security agreement as another reminder of the same. Further information, such as advertising copy and the like, may be provided in a footer block 122. As indicated above, the receipt 110 is produced for both the purchaser 12 and the merchant 14, so each copy may be inscribed with a copy identifier 124. Although specifics of the receipt 110 have been described in detail, it will be appreciated by those having ordinary skill in the art that they are presented by way of example only and not of limitation. Any other suitable form and content of

the receipt 110 may be substituted without departing from the scope of the present invention.").

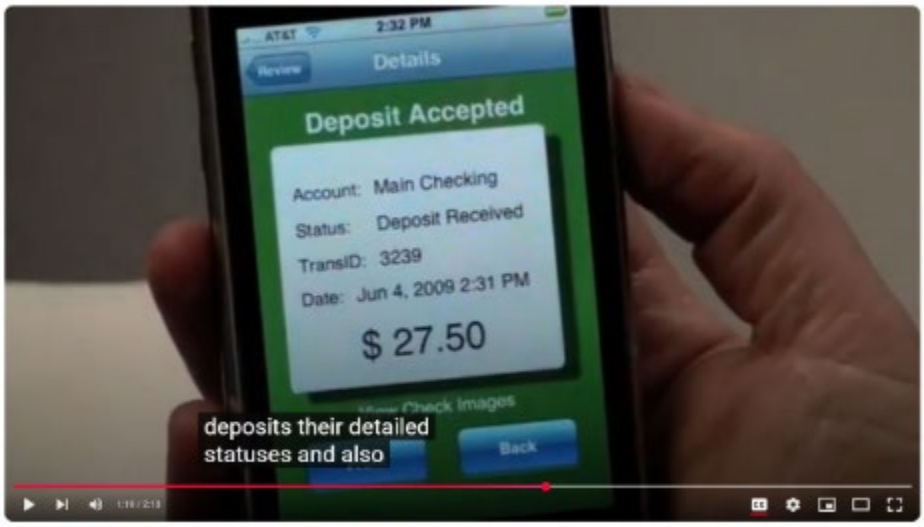
- Viera at 7:33-64 ("FIGS. 5A-E are example graphical user interfaces illustrating a deposit transaction according to another embodiment of the present invention. FIG. 5A illustrates a menu of services 501 that is presented to the user. The user will select a service by touching a labeled button 502. If the user wants to end the transaction, he or she can touch the exit button 503. FIG. 5B illustrates a listing of the user's accounts 504. Each account is displayed with the account type 505, the balance 506, and the available funds 507. The user selects the account that will be receiving the deposit. FIG. 5C illustrates a check deposit 508. A check is inserted into an image scanner without an envelope. The account type 509 is displayed. Verification of check scan 510 is displayed. The user uses a keypad 511 to enter the deposit amount 512. An image 513 of the check is displayed to the user. FIG. 5D illustrates an appraisal of funds availability. The account type 515 is displayed. The deposit amount 516 is also displayed. The funds that are immediately available are displayed at 517A. The availability of the remaining funds is displayed at 517B. The user can reject the deposit by touching button 519 or button 518 to accept. If the deposit is accepted, the deposit information is stored in a remote database that may be accessed by the user's financial institution. For example, after the transaction has completed, FIG. 5E illustrates instant online review of the deposit. A financial institution employee can review the details of the deposit and also see an image of the actual check. Previously, a check must be placed into an envelope for deposit. The envelope then sits in a depository until a teller can process the deposit and determine the availability. A user can then find out what the funds availability is through contacting the financial institution. The institution may also send a letter to the account holder's home with the funds availability.").

- Byrne at [0174]-[0175] ("An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic. The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned."), [0189]-[0190] ("An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent. The acknowledgment is uploaded to the user's client, the content of which depends on the client options.").

- Nepomniachtchi at 9:53–64 ("In some embodiments, the system might determine the amount by processing the image. For example, in some cases, optical character recognition ("OCR") might be used to determine what characters and numbers are present on the document. For example, numbers located in the amount box of a check or payment coupon might then be determined using OCR

or other computer based character determination. This might be done instead of requiring the amount to be entered manually. In other embodiments, a manual entry might be used to verify a computer generated value that is determined using, for example, OCR or other computer based character determination.").

- ImageNet, Mitek - Mobile Online Banking - Depositing your check explained at 1:19.




Mobile Online Banking: Depositing your check *Explained* | Mitek

- WVUFCU (iPhoneApp) at 1; Finovate Blog at 4:

**iPhone app landing page (link (http://www.wvunited.org/iphone.htm))**




(http://www.netbanker.com/WindowsLiveWriter/a40dc3cbb662_7C06/image_8.png)

### n. Updating an Account Balance ('310, '095 Patent)

Before the Asserted Patents, it was well known to update a user's balance based on successful deposit of a check. For example:

- Garcia at 12:22-13:5 ("Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected."), 7:18-8:14 ("This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.").

- Acharya at Claim 20 ("[R]eceiving confirmation that the value of the financial instrument has been deposited into the account of the banking customer by the automated electronic banking system without physical receipt of the financial instrument by the automated electronic banking system."); 3:22–33 ("The automated system can provide immediate provisional credit from the BOFD system to the banking customer, and forward the digital image of the financial instrument and other data to a clearing house in the form of an ECP transaction. Once the transaction has cleared the paying bank, the BOFD system issues a permanent credit to the banking customer. Accordingly, the banking customer has initiated deposit of a financial instrument from a third party, such as a paper check, from the convenience of his/her own home using a digital image of the financial instrument and without having to physically present the financial instrument to the BOFD system, with the speed of an electronic transaction.").

- Lin at [0162] ("In certain embodiments, the approval step may include transmitting the check information to a verification service for approval. For example, the merchant may hire a third party check verification service to verify checks received from customers. In these embodiments, selection of the graphical element 228 may cause the device to send the check information to the external server 96 (FIG. 4), which may in turn send the check information to the check verification service. The check verification service may approve the check for payment and transmit the approval to the electronic device 10 through the external server. Upon receipt, the payment authorization may be displayed within the window 242. The check information also may be transmitted to the customer's bank to verify that the customer has sufficient funds to process the transaction. Further, in other embodiments, the approval process may simply include verifying

that all the check information has been entered, and no additional verification may be performed. After the transaction has been approved, the merchant may select the POP graphical element 224 to conduct another transaction.").

- Rodriguez at [0034] ("Once e-check clearer 142 determines that the funds are available in purchaser's account 144, e-check clearer 142 may transmit a confirmation to POS terminal 130 and transfer the funds to merchant's account 152 via LAN/WAN 160. In other embodiments, purchaser's bank 140 may transfer the funds to merchant's account 152 and merchant's bank 150 may confirm the receipt of the funds with POS terminal 130."); [0006] ("The merchant's bank is disadvantaged because the merchant's bank will show the balance of the check in the merchant's bank account for the 'float' period. Further, the merchant is disadvantaged because the merchant will not know during the delay, possibly for several days, whether the check is actually good, i.e., whether the purchaser actually has sufficient funds available for the transaction, and, even if the purchaser does have the funds available, the funds that show in the merchant's account may be placed on hold by the merchant's bank until they are actually received or for some standard number of days to make sure that the funds are received before being available to the merchant.").

- Fellerman at [0061] ("In further detail, the images of the checks 24 received from the remote terminal 26 or the check scanner 28 as described above are stored at the central payment processor 52. At the specified 30-day intervals, the images of the checks 24 are transmitted as remote deposits under the Check 21 framework into the processor account 108, at which time the bank initiates a request with the purchaser's bank 20 for the transfer of the appropriate amount of funds therefrom. It is also contemplated that third party processors other than the banks may handle the transfers. As previously noted, it is possible for a copy of the check 24 to function as an ACH debit authorization by way of the security agreement. In this regard, the central payment processor 52 is understood to originate an ACH credit thereto that is to be deducted from the purchaser's bank account 20. The ACH credit is likewise originated at 30-day intervals according to the predefined schedule described previously. To the extent that any check 24 fails to clear, the central payment processor 52 is responsible therefor, as the reimbursement obligation to the merchant 14 remains. However, because the central payment processor 52 retains interest in the purchaser's repayment obligations, it may independently initiate a collection action against the purchaser 10 without involvement of the merchant 14."), [0016] ("One alternative to short-term financing with credit cards is a payment structure in which future dated checks are sequentially processed on the dates specified thereby, with representative examples being disclosed in U.S. Pat. App. Pub. No. 2006/0015428 and U.S. Pat. App. Pub. No. 2006/0015427, both to Friedman. In further detail, Friedman contemplates a check writer presenting one or more future-dated checks to a merchant, followed by a verification of the presenter's check writing history. Furthermore, the balance history of the presenter's account may also be evaluated to determine an appropriate monthly payment amount or total financing amount. If these verifications are acceptable, the merchant assigns all of the checks to a

guarantee entity, and immediately receives the total amount for all of the checks presented, minus a service charge. The check guaranty entity then deposits the checks on the dates specified in each of the future-dated checks.").

- Viera at 7:1-32 ("FIG. 4 illustrates a method of performing a transaction on a network according to one embodiment of the present invention. FIG. 4 illustrates a method of performing a deposit transaction. At step 401, a user inserts a check into an image scanner at a kiosk. At step 402, the back and front of the check are scanned and the images are stored (e.g. bitmaps). The magnetic encoding ("MICR") at the bottom of the check is also scanned. The user enters the amount of the deposit using a touchscreen keypad at step 403. At step 404, the bitmap images and MICR obtained from step 402 are passed to an optical character recognition ("OCR") program. OCR will perform several functions. It will re-read the MICR using optical instead of magnetic techniques. It will detect the check amount, payee name, and check number. It will detect the imprinted name of the paying financial institution. It will determine if a signature is present on the front of the check. It will also determine if a signature is present in the endorsement section on the back of the check. A deposit type is determined at step 405 using the information obtained from step 404. In making this determination, risk policies and routing group information from database tables 406 and 407 are used. If deposit is rejected at step 408, the check is returned to the user. If the deposit is accepted, it will be examined for hold override policies using policy information stored in database 411. Funds availability is calculated and presented to the user at step 412. If the user rejects the deposit at step 413, then the check will be returned at step 414. If the user elects to proceed, deposit information will be printed on the check and stored in a secure internal bin at step 415. At step 416, the check images and deposit information will be posted through the shared branching network 417 to be received by the appropriate financial institution."), 7:32-64 ("FIGS. 5A-E are example graphical user interfaces illustrating a deposit transaction according to another embodiment of the present invention. FIG. 5A illustrates a menu of services 501 that is presented to the user. The user will select a service by touching a labeled button 502. If the user wants to end the transaction, he or she can touch the exit button 503. FIG. 5B illustrates a listing of the user's accounts 504. Each account is displayed with the account type 505, the balance 506, and the available funds 507. The user selects the account that will be receiving the deposit. FIG. 5C illustrates a check deposit 508. A check is inserted into an image scanner without an envelope. The account type 509 is displayed. Verification of check scan 510 is displayed. The user uses a keypad 511 to enter the deposit amount 512. An image 513 of the check is displayed to the user. FIG. 5D illustrates an appraisal of funds availability. The account type 515 is displayed. The deposit amount 516 is also displayed. The funds that are immediately available are displayed at 517A. The availability of the remaining funds is displayed at 517B. The user can reject the deposit by touching button 519 or button 518 to accept. If the deposit is accepted, the deposit information is stored in a remote database that may be accessed by the user's financial institution. For example, after the transaction has completed, FIG. 5E illustrates instant online review of the deposit. A financial institution employee can review

the details of the deposit and also see an image of the actual check. Previously, a check must be placed into an envelope for deposit. The envelope then sits in a depository until a teller can process the deposit and determine the availability. A user can then find out what the funds availability is through contacting the financial institution. The institution may also send a letter to the account holder's home with the funds availability.").

- Cohen at [0051] ("Step 12: Upon acceptance of the X937 file by the Federal Reserve Bank, further processing of the checks is performed by the Federal Reserve Bank. The Federal Reserve Bank performs a net settlement of funds to the BOFDs."), [0025] ("The DirectFed™ system takes advantage of the new rules provided by the Check 21 act. It allows merchants to deposit checks without having to deliver the physical check to a bank. The DirectFed™ clearing system has been approved by the Federal Reserve Bank for the delivery of checks via the Check 21 system. The checks, preferably delivered/routed via a third party system, such as the DirectFed™ clearing system 118, are directly deposited with the Federal Reserve Bank 130 (or Endpoint Exchange 132), completely bypassing the BOFD 128. There are a host of benefits and advantages provided by the system and method of the present invention, including, for example, the following: (1) merchant employees do not have to leave the office to deposit checks; (2) manual deposit tickets do not have to be created; (3) photocopies of checks do not have to be created, since the images of the checks are retained and available on the client system; (4) for large volume deposits, armored guards/vehicles are not required, since the physical checks are never transported to the bank; (5) rather than having paper checks delivered to the bank and then manually processed at the bank, the check images are electronically delivered to the Federal Reserve Bank, or Endpoint Exchange, for credit to the BOFD, resulting in greatly expedited availability of funds; (6) due to the increased clearance speed, returns are processed faster, giving merchants a greater opportunity to recover funds; (7) upon request of the merchant, checks can be automatically separated into those eligible for ACH and those which are not eligible, so that ACH eligible checks can be processed by ACH and the others via Check 21, preventing the requirement of the merchant manually separating checks before processing via ACH. With the addition of coupon processing (further described herein below), or without coupon processing (depending upon the required data), automated import into client Accounts Receivable (A/R) and other systems can be easily accommodated. This can result in an enormous saving of time, and elimination of many common keyboard errors. The BOFD does not need to have a physical location near the remote merchant site. This allows a bank to expand the geographic coverage of its depositor. Similarly, a merchant with multiple locations can consolidate its accounts at a single bank, even for those locations that are not located near a bank branch. Banks that clear checks through other banks, or credit unions clearing through corporate credit unions, can bypass the intermediary and clear directly through the Federal Reserve Bank. This saves time and the cost of physically delivering the checks.").

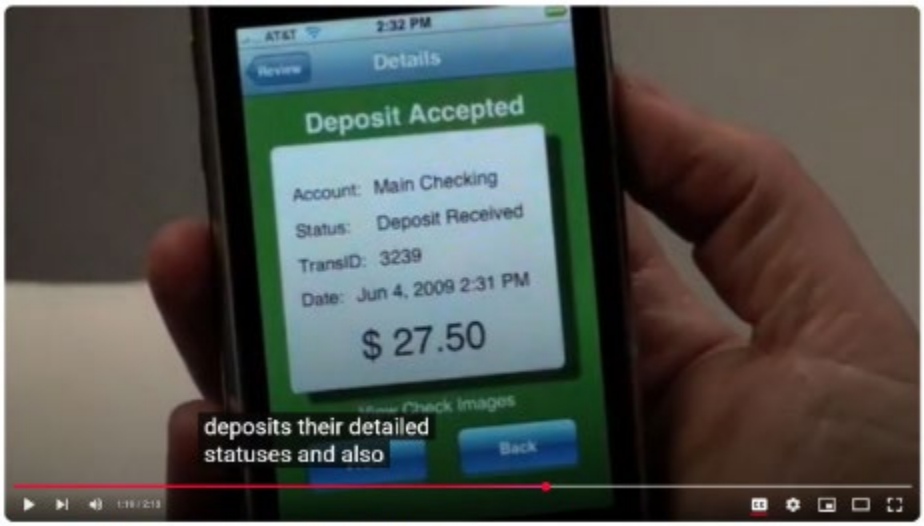
- Byrne at [0014] (“It would be advantageous to improve operating efficiencies by providing faster access to money deposited in a customer’s account and providing increased flexibility to invest and direct funds.”).

- CAM (Parikh 2005a) at 2 (“3. Applications
We believe that CAM is suitable for a rich set of applications in rural India and elsewhere. We have met banks, private companies, NGOs and governmental agencies that have expressed interest in using the CAM framework. Below are some of the applications that we are currently exploring:
- Sales and inventory tracking
- Health data collection
- Transaction processing
- Mobile cash register”).

- CAM (Parikh 2007) at 56 (“Figure 4.2 shows a CAM-enabled loan application designed for a microfinance institution. The barcode in the top right, when clicked, activates a function that displays a sequence of prompts for the user to enter each of the values from the form. A section of the code for this function is shown below. input_date, conf_note, recordaudio and put are all CAM API functions. A full list of API functions is given in Table 4.2.”).

- CAM (Parikh 2007) at 75 (“The resulting reports and financial statements are printed from a secure access point, or faxed from the Federation head office to the field office, where they are picked up by the group or delivered by the staff. Reports can contain interactive CAM content, for example, for accessing up-to-date account balances. These reports are filed in a set of folders, taking the place of the shared notebooks. They are used by the group for monitoring their portfolio and applying for loans and other services. Paper remains the main display and storage medium for SHGs and their members.”).

- ImageNet, Mitek - Mobile Online Banking - Depositing your check explained at 1:19.




### o. Converting the Format of a Check Image ('310, '095 Patent)

Before the Asserted Patents, it was well known to convert the format of an image of a check for deposit, including to convert the format to comply with ANSI X9.37 DSTU—2003. For example:

- Cohen at [0026] ("In contrast to systems and methods of the prior art, the system and methods of the present invention enable and facilitate deliver directly to the Federal Reserve Bank. Existing systems require delivery to the BOFD, which then processes the checks for clearance. Furthermore, the system and methods of the present invention enable automatic determination as to whether to process a check by ACH or Check 21. The DirectFed™ clearing system is capable of creating multiple Check 21 compliant X937 files, each file containing multiple cash letters and bundles. This allows delivery of checks for multiple banks to multiple Federal Reserve Banks for multiple accounts."), claim 1 ("a second subsystem at said client location, in secure communication with said first subsystem, configured for receiving checking document data, converting the checking document data into a format in conformance with Check 21 processing requirements, and securely transmitting said formatted data to said first subsystem for direct routing to said Federal Reserve bank.").

- Acharya at 8:58–9:10 ("Next, the BOFD system 110 may make a decision concerning the method for "clearing" the financial instrument through another bank 345. If the transaction will be converted to an Electronic Funds Transfer (EFT), the BOFD system 110 may have to obtain a payor authorization 350 to support preparation of the EFT transaction 355 using the financial instrument data that was transmitted from the RCT 100 to the BOFD system 110. Alternatively, the BOFD system 110 may format the transaction for Electronic Check

Presentment (ECP) 360 using the financial instrument data that was transmitted from the RCT 100 to the BOFD system 110, according to certain pre-established agreements with the paying bank. If, however, the BOFD system 110 does not have either the payor's authorization for an EFT, or an agreement with the paying bank for ECP, then the BOFD may print a reconstructed check 365 using the financial instrument data that was transmitted from the RCT 100 to the BOFD system 110. In any of these alternatives, the deposit of the financial instrument can be "cleared" without physical receipt of the financial instrument by the BOFD system 110.").

- Rodriguez at [0049] ("Considering FIG. 2 in conjunction with FIG. 1, there is shown an example e-check 200. E-check 200 may include a digital file that includes the same or similar data as data included on a paper check. E-check 200 may include, e.g., payer 210 and payee 230. Payer 210 includes a routing number 212 that identifies the purchaser's bank 140, an account number 214 that identifies the purchaser's account 144 at the purchaser's bank 140 from which funds in an amount 218 may be withdrawn, a check number 216, the amount 218, and a check image 220. Routing number 212 may include an electronic address like an IP address 240 and a port number 242, determined from, e.g., a routing number table, to describe the electronic address for the purchaser's bank that is utilized for clearing checks. Further, check image 220 may include an image in a standard digital format such as Joint Photographic Experts Group (JPEG), Graphics Interchange Format (GIF), Tagged Image File Format (TIFF), Portable Network Graphics (PNG), or other graphics format. Thus, upon receipt and processing of e-check 200, purchaser's bank 140 may link check image 220 to a transaction described in an e-statement available for purchaser's account 144.").

- Fellerman at [0037] ("The merchant 14 may operate a remote terminal 26 that includes a check scanner 28 that is capable of capturing the images of the checks 24. In capturing the images, for example, a Tagged Image File Format (TIFF) file may be created. Any one of widely available check scanners 28 that can be controlled from the remote terminal 26 may be utilized, such as the EC5000 series and EC6000 series imagers from RDM Corporation of Waterloo, Ontario, Canada."), [0038] ("FIGS. 4 a and 4 b are representative examples of the front and back, respectively, of the check 24 and it is understood that the check scanner 28 produces data representative thereof for electronic transmission. Preferably, the format of the data produced by the check scanner 28 conforms to the X9.90 standard that sets forth the construction, layout, data elements, data content, and printing specifications for the truncated check 24.").

- Byrne at [0123] ("The application uses a database, such as Oracle, for storing information. The images local to the system can be stored in Binary Large Object (BLOB) format"); [0087] ("The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique

item sequence number (ISN), based on Wells Fargo standards and including a parameter driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to Support easy retrieval of the original item from temporary storage when necessary."); [0088] ("This file format is based on the ANSI X9.37 DSTU 2003 "Specifications for Electronic Exchange of Check and Image Data" draft standard with some modifications to accommodate the processing of credits, such as customer Direct Deposit Account (DDA) deposits and adjustments, and to augment sender Source identification and file balancing controls.").

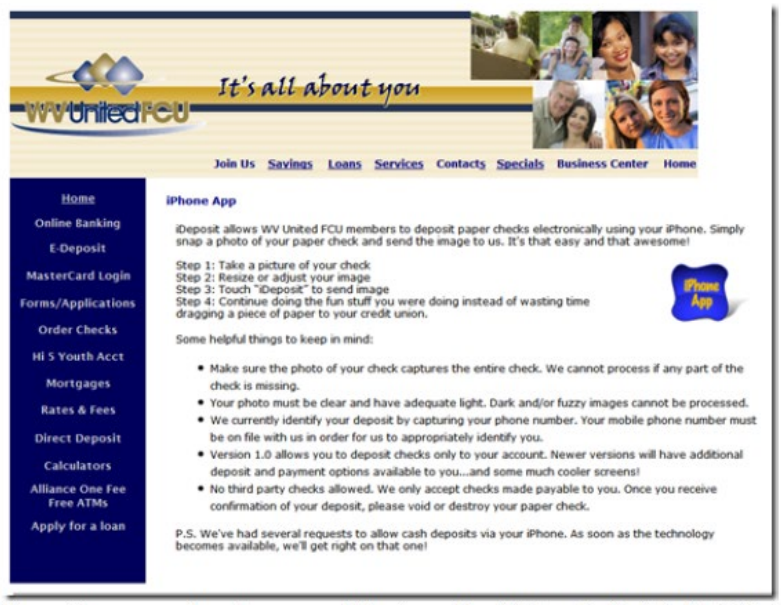
- Byrne at [0088] ("This file format is based on the ANSI X9.37 DSTU—2003 "Specifications for Electronic Exchange of Check and Image Data" draft standard with some modifications to accommodate the processing of credits, such as customer Direct Deposit Account (DDA) deposits and adjustments, and to augment sender source identification and file balancing controls.").

- Nepomniachtchi at Abstract ("The present invention relates to automated document processing and more particularly, to methods and systems for document image capture and processing using mobile devices. In accordance with various embodiments, methods and systems for document image capture on a mobile communication device are provided such that the image is optimized and enhanced for data extraction from the document as depicted. These methods and systems may comprise capturing an image of a document using a mobile communication device; submitting the image to a server; and processing the image to create a bi-tonal image of the document for data extraction. Additionally, these methods and systems may comprise capturing a first image of a document using the mobile communication device; automatically detecting the document within the image; geometrically correcting the image; binarizing the image; correcting the orientation of the image; correcting the size of the image; and outputting the resulting image of the document."); 1:56–2:3 ("The present invention relates to automated document processing and more particularly, to methods and systems for document image capture and processing using mobile devices. In accordance with various embodiments, methods and systems for document image capture on a mobile communication device are provided such that the image is optimized and enhanced for data extraction from the document as depicted. These methods and systems may comprise capturing an image of a document using a mobile communication device; transmitting the image to a server; and processing the image to create a bi-tonal image of the document for data extraction. For example, a mobile communication devices, such as a camera phone, would transmit the image of the document to the server, where the image is processed and results in a bi-tonal image of the document."); 4:65–5:7 ("The present invention is directed towards automated document processing and systems and methods for document image processing using mobile devices. Generally, some embodiments of the invention capture an original color image of a document using a mobile device and then converted the color image to a bi-tonal image. More specifically, some embodiments accept a color image of a check taken by a mobile device and convert it into a bi-tonal image of the check that is

substantially equivalent in its resolution, size, and quality to check images produced by "standard" scanners"); 10:10–17 ("In other embodiments, some or all data processing might be performed at the mobile communication device. For example, the mobile communication device might perform auto-rotate, de-skew, perspective distortion correction, cropping, etc. Additionally, the mobile device might also process the image to produce a bi-tonal image for data extraction. In some cases, the processing might be shared between the mobile device and the server."); 19:64–67 ("AW is the average width of the MICR-character found; and $AW_{200}$ is the corresponding "theoretical" value based on the ANSI×9.37 standard (Specifications for Electronic Exchange of Check and Image Data) at 200 DPI.").

- WVUFCU (iPhoneApp) at 1; Finovate Blog at 4:

**iPhone app landing page (link (http://www.wvunited.org/iphone.htm))**




(http://www.netbanker.com/WindowsLiveWriter/a40dc3cbb662_7C06/image_8.png)

### 2. Obviousness Combinations

The accompanying claim charts identify how each prior art reference discloses the limitations of the asserted claims on a limitation-by-limitation basis, and illustrative combinations are identified below. If USAA argues that any particular prior art reference lacks any feature for which no combining references are provided in the relevant claim chart, a person of ordinary skill in the art as of the Asserted Patents' purported invention date(s) would at a minimum have been motivated to modify the reference to include the allegedly missing feature,

or to combine it with other references that include that feature, as discussed in Section V.B.3 below.

Illustrative obviousness combinations are listed below (it should be understood that each listed combination can also include the general knowledge of a person of ordinary skill in the art at the time of the Asserted Patents' priority date(s)):

**a. '310 Patent**

| '310 patent claim(s) | Obviousness Combination(s) |
|---|---|
| All | • Garcia in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)<br>• Acharya in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)<br>• Nepomniachtchi in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)<br>• Lin in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)<br>• Rodriguez in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)<br>• ImageNet in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)<br>• WVUFCU in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, RedLaser, or ScanZoom) |
| 13 | • Garcia in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, |



USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 94 of 1772

| '310 patent claim(s) | Obviousness Combination(s) |
|---|---|
| | Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom) and further in view of Yoon<br>• Acharya in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)and further in view of Yoon<br>• Nepomniachtchi in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom) and further in view of Yoon<br>• Lin in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom) and further in view of Yoon<br>• Rodriguez in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom) and further in view of Yoon<br>• ImageNet in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom) and further in view of Yoon<br>• WVUFCU in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, RedLaser, or ScanZoom) and further in view of Yoon |

**b. '095 Patent**

| '095 patent claim(s) | Obviousness Combination(s) |
|---|---|
| All | • Garcia in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)<br>• Acharya in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), |



USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 95 of 1772

| '095 patent claim(s) | Obviousness Combination(s) |
|---|---|
| | and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)<br>• Nepomniachtchi in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)<br>• Lin in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)<br>• Rodriguez in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)<br>• ImageNet in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)<br>• WVUFCU in view of (one of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, RedLaser, or ScanZoom) |
| 5, 13, 22 | • Garcia in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom) and further in view of Yoon<br>• Acharya in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom)and further in view of Yoon<br>• Nepomniachtchi in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom) and further in view of Yoon<br>• Lin in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom) and further in view of Yoon<br>• Rodriguez in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or |



USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 96 of 1772

| '095 patent claim(s) | Obviousness Combination(s) |
|---|---|
| | (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom) and further in view of Yoon<br>• ImageNet in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, WVUFCU, RedLaser, or ScanZoom) and further in view of Yoon<br>• WVUFCU in view of (one of Luo, Zhu, CAM, MotoRokr, RedLaser, or ScanZoom), (one of Cohen, Fellerman, Viera, or Byrne), and/or (one of Meier, Ranjan, Android Developers Website, Android Website, Asai, Goyal, CAM, RedLaser, or ScanZoom) and further in view of Yoon |

### 3. Reasons to Modify, Extend, or Combine the Prior Art References

As set forth in Sections V.A and V.B and Exhibits A1 through B24, each and every limitation of the Asserted Claims of the Asserted Patents is disclosed and/or rendered obvious at least when applying USAA's apparent interpretation of the claims in its P.R. 3-1 and 3-2 disclosures. The Asserted Patents do not introduce any new concepts for mobile devices or check deposit, nor do the claims combine these known concepts in any inventive or non-obvious manner.

The Supreme Court has held that "[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 416 (2007). "When a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one." *Id.* at 417. As the Supreme Court made clear, "[f]or the same reason, if a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill." *Id.*



USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 97 of 1772

In order to determine whether there is an apparent reason to combine the known elements in the fashion claimed by the patents at issue, a court can "look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art." *Id.* at 418. For example, obviousness can be demonstrated by showing "there existed at the time of invention a known problem for which there was an obvious solution encompassed by the patent's claims." *Id.* at 420. "[A]ny need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed." *Id.* Common sense also teaches that "familiar items may have obvious uses beyond their primary purposes, and in many cases a person of ordinary skill will be able to fit the teachings of multiple patents together like pieces of a puzzle." *Id.*

Thus, the motivation to combine the teachings of the prior art references disclosed herein is found in the references themselves and/or: (1) the nature of the problem being solved; (2) the express, implied, and inherent teachings of the prior art; (3) the knowledge of persons of ordinary skill in the art; (4) the fact that the prior art is generally directed towards the same or similar problems; and/or (5) the predictable results obtained in combining the different elements of the prior art.

The foregoing general comments are relevant to how a person of ordinary skill in the art ("POSITA") would understand prior art references in the field(s) of the Asserted Patents, and also relevant to which concepts a person of ordinary skill would deem obvious—both for single-reference obviousness and obviousness combinations. Patents related and covering similar concepts to the Asserted Patents were challenged in various *Inter Partes* review proceedings before the Patent Trial and Appeal Board ("PTAB"). The PTAB issued Final Written Decisions

in five of proceedings finding challenged claims obvious under 35 U.S.C. § 103 over prior art cited herein:

- *PNC Bank, NA v. United Services Automobile Association*, Case No. IPR2021-01070, Paper 68 (January 19, 2023) (finding all challenged claims of U.S. Patent No. 8,699,779 related to the '310 and '095 patents invalid over Acharya and Luo).
- *PNC Bank, NA v. United Services Automobile Association*, Case No. IPR2021-01073, Paper 72 (January 19, 2023) (finding all challenged claims of U.S. Patent No. 8,977,571 related to the '310 and '095 patents invalid over Acharya, Luo, Nepomniachtchi, and Yoon).
- *PNC Bank, NA v. United Services Automobile Association*, Case No. IPR2021-01077, Paper 73 (January 20, 2023) (finding all challenged claims of U.S. Patent No. 10,621,559 invalid over Garcia, Randle,[5] Slater,[6] Lev,[7] Watanabe,[8] Aoyama,[9] and Byrne).
- *PNC Bank, NA v. United Services Automobile Association*, Case No. IPR2022-00076, Paper 47 (May 10, 2023) (finding various challenged claims of U.S. Patent No. 10,769,598 invalid over Garcia, Byrne, Singfield,[10] and Adusumilli[11]).
- *PNC Bank, NA v. United Services Automobile Association*, Case No. IPR2022-00075, Paper 47 (June 9, 2023) (finding all challenged claims of U.S. patent No.

[5] U.S. Patent Application Publication No. 2006/0106717.
[6] European Patent Application Publication No. 0984410.
[7] U.S. Patent Publication No. 2006/0164682.
[8] U.S. Patent No. 7,027,171.
[9] U.S. Patent Publication No. 2002/0172516.
[10] U.S. Patent Application Publication No. 2005/0097046.
[11] U.S. Patent Application Publication No. 2003/0097592.

9,224,136 invalid over Garcia, Acharya,[12] Richardson,[13] Randle, Slater, Pintsov,[14] and Davis[15]).

The Court of Appeals for the Federal Circuit affirmed all findings of obviousness, and reversed the PTAB's determination that a subset of the dependent claims of the '598 patent were not invalid. Thus, all challenged claims were ultimately found invalid. In view of those decisions, USAA cannot purport to have invented mobile remote deposit capture or automatic capture of check images. To the extent that the claims of these patents recite anything more, the claims are merely trivial and obvious variants of what the PTAB and Federal Circuit already found obvious. Regions reserves the right to rely on the filings and decisions in these IPRs as well as the prior art cited therein as showing what was known and obvious before the '310 and '095 patents.

**a. The '310 and '095 Patents**

i. <u>It would have been obvious to a POSITA to provide an alignment guide ('310 and '095 Patents)</u>

To the extent USAA contends that the prior art identified in Section V.A does not disclose an alignment guide, a POSITA would have found such feature obvious in view of Luo, Zhu, Yoon, and/or Lin, at least under USAA's interpretation of the term. As an illustrative, non-limiting example, a person of ordinary skill in the art would have been motivated to combine the teachings of references such as Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (which disclose, at least in one embodiment, depositing a check using an image captured by a mobile device) with references such as Luo, Zhu, Yoon, CAM, MotoRokr, RL,

[12] International Patent Application No. WO 01/61436.
[13] U.S. Patent Application Publication No. 2005/0281450.
[14] U.S. Patent Application Publication No. 2007/0118747.
[15] U.S. Patent Application Publication No. 2005/0216410.

ScanZoom, and/or Lin (which disclose an alignment guide) to produce a higher quality image. Such a combination would render obvious, for example, limitations containing the phrase "overlaying an alignment guide on the live video of the check in the field of view of the camera" in claims 1, 2, 5, 6, 8, and 9 of the '310 patent and claims 1 and 9 of the '095 patent.

A person of ordinary skill in the art would have been motivated to combine such references because the references are all in the same field (image capture and processing) and address the same problem or objective (capturing images of sufficient quality for future processing) in similar ways (with user devices). *See e.g.*, Yoon at Abstract ("Provided are an apparatus and method for photographing a business card in a portable terminal. The method includes recognizing boundary lines of the business card from an image received from a camera when a recognition and photographing key for a business card is selected, and allowing the business card to be automatically photographed by using the recognized boundary lines."); Luo at Abstract ("A method and system for capturing the frontal image of an object having at least two substantially straight edges (205), which helps reduce projective distortion in the image."); Zhu at 7 ("An object of the present invention is to introduce a method of scanning business cards using a mobile phone with a digital camera. The method is implemented on a mobile phone having a digital camera and involves a business card scanning feature, but does not have the drawbacks of the background art."); Lin at [0160] ("The check information also may be acquired through the camera. For example, the graphical element 236 may be selected to enter the check information using techniques similar to those described with respect to FIGS. 13 and 14."), [0162] ("In certain embodiments, the approval step may include transmitting the check information to a verification service for approval."); CAM (Parikh 2005a) at 1 ("In this paper we introduce mobile document processing as an accessible and locally harmonious way to provide

information services to the developing world [2]. In our system a camera-equipped mobile phone is used for image capture and data entry. Document interaction is specified using CamShell - a narrative document-embedded programming language that can allow developers to add interactive audio, data entry, validation, processing, and networking instructions to otherwise inert paper documents."); MotoRokr (MotoRokr Manual) at 108, 98 ("Your phone automatically captures a business card image when the card is properly framed in the OCR viewfinder. (To activate or deactivate the auto-capture feature, see "business card reader settings" on page 97)."); RedLaser (TechCrunch) at 1 ("Now that the new iPhone 3G S has a video camera, TechStars startup Occipital has released an update to its RedLaser app (iTunes link, $1.99) which speeds up barcode scanning by not requiring you to first take a picture. Occipital claims that its 'realtime barcode scanner' is the only one which works on phones with both autofocus (the new 3G S) and without (the older iPhone and iPod Touch). Other barcode scanning apps, such as ShopSavvy's, can also take advantage of the autofocus camera on the 3G S, but can't do on-the-fly scanning on the older models."); ScanZoom (Attia) at 3:45-52 ("3:45-52 ("The user then takes a picture of the barcode using the digital camera. As soon as the barcode is captured, the ScanZoom software decodes the barcode utilizing a decoding engine integral to the ScanZoom software. Alternatively, the Scanzoom software may send the image of the barcode to a central server for decoding by sending a multimedia message service ("MMS") message to the server containing the barcode image.").

As explained in Section V.B.1.i, it was well known to display an alignment guide on a mobile device screen when capturing an image of a document to improve the quality of the resulting image. *See, e.g.*, Luo at Abstract ("A method and system for capturing the frontal image of an object having at least two substantially straight edges (205), which helps reduce

projective distortion in the image. The method includes receiving a preview image on an image sensor (115) related to a camera (100), and the preview image depicts a straight edge (205). Then, a preview image depicting a straight edge (205) and at least two reference lines (135) simultaneously present is displayed in a preview window (125) of the camera system (100), where each reference line (135) corresponds to a straight edge (205). The image of the object is captured only when the straight edge (205) shown in the preview window (125) is substantially parallel to the corresponding reference line (135).”); Yoon at [0019] (“FIG. 1 is a block diagram of a portable terminal according to the present invention. Referring to FIG. 1, the portable terminal includes a controller 100, a camera unit 102, an image processor 104, and a display unit 106….Also, the controller 100 displays on the display unit 106 reference boundary lines, for presenting a size and location, of a business card, which are appropriate for recognizing the business card; and determines whether the boundary lines of the business card received from the image processor 104 coincide with the reference boundary lines. If they coincide with one another, the controller 100 controls automatic photographing to be performed.”); Zhu at 7 (“To achieve the object described above, the present invention employs the technical solution below. While retaining steps (2), (3), and (4) of the background art above, it improves step (1) by: detecting the four sides of the targeted business card on the mobile phone screen, drawing the detected four sides of the targeted business card using lines of a first color so that the mobile phone user can clearly see the position and shape of the business card on the screen, and using arrows based on the current position of the business card to prompt the mobile phone user to move the targeted business card in the direction pointed to by the arrows until the business card arrives at a position suitable for photographing; when the relative positions of the mobile phone digital camera and the business card are suitable for photographing, the four sides of said

business card become a second color, prompting the mobile phone user to prepare to take a photograph; to photograph the targeted business card, the mobile phone user needs only to hold the mobile phone steady and trigger the "shoot" button of the digital camera, either manually or automatically by the mobile phone, thus overcoming a drawback, namely the tendency of the camera to shake during shooting."); Lin at [0154] ("Instructions 304 prompt the merchant to take a picture of the credit card. The merchant may position the electronic device 10 so that the card image is aligned within the frame 302, and then the merchant may select graphical element 306 to capture the image. If the merchant would like to recapture the image, the merchant may select the graphical element 260 to return to the previous screen 218. If the merchant has successfully captured the image, the merchant may select the graphical element 228 to begin the payment process using the captured image. Selection of the graphical element 228 may cause a CVV entry screen 308 to be displayed. Similar to the process described with respect to FIG. 11, the merchant may enter the CVV using the keypad 266. After entry, the merchant may select the pay button 274 to process the payment. The payment status screen 240 may then be displayed and an approval message may be displayed within the window 242 to notify the merchant when the transaction has been approved."), [0160] ("The check information also may be acquired through the camera. For example, the graphical element 236 may be selected to enter the check information using techniques similar to those described with respect to FIGS. 13 and 14. In certain embodiments, the check information may be acquired using the scanner 48 (FIG. 2). For example, the graphical element 336 may be selected to scan a bar code on the check using the scan device 48 (FIG. 2). In other embodiments, the payment information may be acquired by scanning an NFC enabled check. In these embodiments, a graphical element may be included within the summary bar 152 to enter the check information using near field communication. The

check information may be acquired by near field communication using techniques similar to those described above with respect to FIGS. 11-12."); CAM (Parikh 2007) at 53-54 ("For barcode recognition and generation, I am using the open source toolkit developed by Rohs and Gfeller [128]. This includes a 2D barcode recognition library for Nokia Series 60 phones. A user can capture a barcode by taking a picture of it using the mobile phone [Figure 4.3] camera. Each barcode contains 83 bits of data, 7 of which are being used for error correction (leaving 76 bits of data capacity). Several barcodes can be detected from a single camera image."); MotoRokr (MotoRokr Manual) at 108, 98 ("Your phone automatically captures a business card image when the card is properly framed in the OCR viewfinder. (To activate or deactivate the auto-capture feature, see "business card reader settings" on page 97)."); RedLaser (Realtime Video) at 0:01-0:09; ScanZoom (Attia) at 6:51-7:6 ("Referring next to FIG. 2A, shown is the process utilized by ScanZoom to acquire and decode barcode 103 once the application has been launched. A user first launches the ScanZoom program on mobile device 105 by selecting its icon from a list of programs located on the device in step 201. Generally, this will initialize the camera and all the network resources needed by ScanZoom for the application to function properly. Once the user has started the program, a message appears directing the user to take a picture of barcode 103 in step 203. ScanZoom provides the user with a "preview" window which functions as a viewfinder, allowing the user to properly center and align the barcode before taking a picture. Once the barcode image has been acquired by digital camera 107, ScanZoom attempts to decode the barcode utilizing its proprietary barcode decoding algorithm in step 205. If the software cannot decode barcode 103 on a first attempt in step 205, the ScanZoom software attempts to decode the image a finite amount of times. The number of decoding attempts can either be fixed or able to be altered by the user. Once barcode 103 has been properly decoded as verified in step

207, mobile device 105 prepares the decoded information for transfer to server 113 to undergo further processing in step 209.").

A person of ordinary skill in the art would have been motivated to incorporate these teachings into Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU because doing so would help a user properly position a check in a preview window before capturing an image thereof. References such as Luo, Zhu, Yoon, MotoRokr, ScanZoom, and/or Lin disclose the benefits of incorporating alignment guides into such systems. *See, e.g.*, Luo at 5 ("The present invention ensures that the front of the object being imaged is substantially parallel to the image plane 320 of the camera system 100 to reduce the projective distortion of the image."); Lin at [0154] ("Instructions 304 prompt the merchant to take a picture of the credit card. The merchant may position the electronic device 10 so that the card image is aligned within the frame 302, and then the merchant may select graphical element 306 to capture the image. If the merchant would like to recapture the image, the merchant may select the graphical element 260 to return to the previous screen 218. If the merchant has successfully captured the image, the merchant may select the graphical element 228 to begin the payment process using the captured image. Selection of the graphical element 228 may cause a CVV entry screen 308 to be displayed. Similar to the process described with respect to FIG. 11, the merchant may enter the CVV using the keypad 266. After entry, the merchant may select the pay button 274 to process the payment. The payment status screen 240 may then be displayed and an approval message may be displayed within the window 242 to notify the merchant when the transaction has been approved."); Zhu at 5 ("The present invention has the following advantages over the background art: 1. It uses screen prompts to guide the mobile phone user to take high-quality photographs of targeted business cards. The four sides of the targeted business card detected on said screen

enable the mobile phone user to clearly view the current position and shape of the targeted business card, and the prompt arrows can let the mobile phone user clearly know how to move the targeted business card in order to take an acceptable photograph. Screen prompts guide any mobile phone user who is not adept at taking photographs to take high-quality photographs of targeted business cards. 2. As a result of the automatic shooting feature, the mobile phone camera photographs targeted business cards without shaking and thereby helps to improve the quality of photographs taken of targeted business cards. The above two advantages ensure that said mobile phone can take targeted business card photographs that are high-quality and do not contain any "noise," thus ensuring the recognition accuracy of said business card recognition system and increasing the usefulness and convenience of said business card recognition system."); Yoon at [0007] ("In the business card recognition function, the probability of satisfactorily recognizing the business card in order to obtain the information contained in the business card varies depending on photographing conditions, such as the size, brightness, and image quality of the photographed business card. Thus, as described above, when the user photographs the business card based on his/her subjective judgment, the business card may be photographed to be too large, small or dark, thereby lowering the probability of satisfactorily recognizing the business card. Also, as described above, when the user manually selects a key button or a menu in order to photograph the business card, shaking may occur due to the selecting of the key button or the menu, and thus, shaking noise may be present in an image of the photographed business card, thereby significantly lowering the probability of satisfactorily recognizing the business card."); MotoRokr (MotoRokr Manual) at 108, 98 ("Your phone automatically captures a business card image when the card is properly framed in the OCR viewfinder. (To activate or deactivate the auto-capture feature, see "business card reader

settings" on page 97)."); ScanZoom (Attia) at 6:51-7:6 ("Referring next to FIG. 2A, shown is the process utilized by ScanZoom to acquire and decode barcode 103 once the application has been launched. A user first launches the ScanZoom program on mobile device 105 by selecting its icon from a list of programs located on the device in step 201. Generally, this will initialize the camera and all the network resources needed by ScanZoom for the application to function properly. Once the user has started the program, a message appears directing the user to take a picture of barcode 103 in step 203. ScanZoom provides the user with a "preview" window which functions as a viewfinder, allowing the user to properly center and align the barcode before taking a picture. Once the barcode image has been acquired by digital camera 107, ScanZoom attempts to decode the barcode utilizing its proprietary barcode decoding algorithm in step 205. If the software cannot decode barcode 103 on a first attempt in step 205, the ScanZoom software attempts to decode the image a finite amount of times. The number of decoding attempts can either be fixed or able to be altered by the user. Once barcode 103 has been properly decoded as verified in step 207, mobile device 105 prepares the decoded information for transfer to server 113 to undergo further processing in step 209.").

It would have therefore been obvious to a person of ordinary skill in the art to incorporate an alignment guide, for example as taught in Luo, Zhu, Yoon, CAM, MotoRokr, RL, ScanZoom, and/or Lin. Doing so would have involved simply combining prior art elements/techniques (e.g., Luo, Zhu, Yoon, CAM, MotoRokr, RL, ScanZoom, and/or Lin's alignment guide) according to known devices/methods to yield predicable results (e.g., helping capture higher quality images of a document) and the simple substitution of one known element (e.g., one image capture technique) for another (e.g., an image capture technique that involves the use of an alignment guide) to obtain predictable results (e.g., helping capture higher quality images of a document).

A person of ordinary skill in the art would have had a reasonable expectation of success. For example, providing an alignment guide would have simply involved writing computer code to display the image of a guide or other reference lines on the preview window during camera capture. Moreover, as discussed above, Luo, Zhu, Yoon, CAM, MotoRokr, RL, ScanZoom, and/or Lin each teach that the predictable result of using an alignment guide to help capture an image of a document is an image that is less prone to problems like projective distortion and other image quality problems. The references use similar well-known components such that combining the references to add an alignment guide in the proposed combination(s) would have been a routine, predictable design for a person of ordinary skill in the art and would not have involved any undue experimentation.

ii. It would have been obvious to a POSITA to automatically capture an image based on one or more monitoring criteria ('310 and '095 Patents)

To the extent USAA contends that the prior art identified in Section V.A does not disclose automatically capturing an image based on one or more monitoring criteria, a POSITA would have found such feature obvious in view of Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, and/or ScanZoom, at least under USAA's interpretation of the term. As an illustrative, non-limiting example, a person of ordinary skill in the art would have been motivated to combine the teachings of references such as Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (which disclose, at least in one embodiment, depositing a check using an image captured by a mobile device) with references such as Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, and/or ScanZoom (which disclose automatically capturing an image based on a variety of monitoring criteria) to produce a higher quality image. Such a combination would render obvious, for example, all limitations containing the term "monitoring criterion" in claims 1-16 of the '310 patent and 1-5, 9-13, 18-22, and 26-29 of the '095 patent.

A person of ordinary skill in the art would have been motivated to combine such references because the references are all in the same field (image capture and processing) and address the same problem or objective (capturing images of sufficient quality for future processing) in similar ways (with user devices). *See e.g.*, Yoon at Abstract ("Provided are an apparatus and method for photographing a business card in a portable terminal. The method includes recognizing boundary lines of the business card from an image received from a camera when a recognition and photographing key for a business card is selected, and allowing the business card to be automatically photographed by using the recognized boundary lines."); Luo at Abstract ("A method and system for capturing the frontal image of an object having at least two substantially straight edges (205), which helps reduce projective distortion in the image."); Zhu at 7 ("An object of the present invention is to introduce a method of scanning business cards using a mobile phone with a digital camera. The method is implemented on a mobile phone having a digital camera and involves a business card scanning feature, but does not have the drawbacks of the background art."); CAM (Parikh 2005a) at 1 ("In this paper we introduce mobile document processing as an accessible and locally harmonious way to provide information services to the developing world [2]. In our system a camera-equipped mobile phone is used for image capture and data entry. Document interaction is specified using CamShell - a narrative document-embedded programming language that can allow developers to add interactive audio, data entry, validation, processing, and networking instructions to otherwise inert paper documents."); MotoRokr (MotoRokr Manual) at 108, 98 ("Your phone automatically captures a business card image when the card is properly framed in the OCR viewfinder. (To activate or deactivate the auto-capture feature, see "business card reader settings" on page 97)."); RedLaser (TechCrunch) at 1 ("Now that the new iPhone 3G S has a video camera, TechStars startup

Occipital has released an update to its RedLaser app (iTunes link, $1.99) which speeds up barcode scanning by not requiring you to first take a picture. Occipital claims that its 'realtime barcode scanner' is the only one which works on phones with both autofocus (the new 3G S) and without (the older iPhone and iPod Touch). Other barcode scanning apps, such as ShopSavvy's, can also take advantage of the autofocus camera on the 3G S, but can't do on-the-fly scanning on the older models."); ScanZoom (Attia) at 3:45-52 ("3:45-52 ("The user then takes a picture of the barcode using the digital camera. As soon as the barcode is captured, the ScanZoom software decodes the barcode utilizing a decoding engine integral to the ScanZoom software. Alternatively, the Scanzoom software may send the image of the barcode to a central server for decoding by sending a multimedia message service ("MMS") message to the server containing the barcode image.").

As explained in Section V.B.1.j, it was well known to automatically capture an image of a document using a mobile device once the image in a preview window satisfies one or more monitoring criteria to obtain a higher quality image of the document. *See, e.g.*, Luo at 6 ("As an alternative, when the straight edge 205 displayed on the preview window 125 is substantially parallel to the corresponding reference line 135, the system 100 can automatically capture the selected image and provide the user with or without instructions."); Yoon at [0028] ("Next, the portable terminal determines whether all the upper, lower, left and right reference boundary lines coincide with the boundary lines of the business card in step 215. If all the reference boundary lines do not coincide with all the boundary lines of the business card, the portable terminal returns to step 209. If all the reference boundary lines completely coincide with the boundary lines of the business card as illustrated in FIG. 3D, the portable terminal proceeds to step 217 and displays a message informing the user of the start of automatic photographing for the business

card. Here, it is possible to inform the user of the start of automatic photographing by generating a predetermined alarm sound of automatic photographing or changing the shapes, thicknesses or colors of the reference boundary lines."); Zhu at 7 ("…to photograph the targeted business card, the mobile phone user needs only to hold the mobile phone steady and trigger the 'shoot' button of the digital camera, either manually or automatically by the mobile phone, thus overcoming a drawback, namely the tendency of the camera to shake during shooting."); Zhu at 7-8 ("A method of scanning business cards using a mobile phone with a digital camera, comprising four steps: (1) using the built-in digital camera of the mobile phone to photograph the targeted business card; (2) using OCR technology to automatically recognize text in the photograph taken of the targeted business card in step (1); (3) using text classification technology to automatically locate useful information, such as name, address, postal code, telephone number, fax number, and email address, within the text recognized in step (2) and displaying such as recognition results on said mobile phone screen; (4) the mobile phone automatically adding the confirmed recognition results to the address book of said mobile phone after the mobile phone user has checked, corrected, and confirmed the recognition results; the method needs to be implemented on a mobile phone that has a digital camera, said mobile phone having both a camera system containing a single-chip microcomputer system and a business card recognition system, wherein the memory of said single-chip microcomputer system stores programs for controlling and commanding the CPU to execute the following operations to ensure that the method can be implemented normally:…seventh step: said camera system automatically triggering the 'shoot' button of said mobile phone to obtain an image of the targeted business card."); CAM (Pervasive Computing) at 76 ("The CamBrowser application's primary purpose is to capture a visual image of the form and decode and interpret the embedded CamShell program, which executes the user

interface that collects information from the user. CamBrowser executes instructions in sequence, starting with instruction zero. Internally, a simple virtual machine keeps track of the current program counter and local variables. You can scan codes either by pointing the camera at the CamForm (scan mode) or by explicitly taking a picture (click mode). We've found the former to be more suitable for quick, directed user interaction, while the latter is more effective for capturing an entire CamShell program and entering large amounts of data—for example, in a loop."); MotoRokr (MotoRokr Manual) at 108, 98 ("Your phone automatically captures a business card image when the card is properly framed in the OCR viewfinder. (To activate or deactivate the auto-capture feature, see "business card reader settings" on page 97)."); RedLaser (TechCrunch) at 1 ("Now that the new iPhone 3G S has a video camera, TechStars startup Occipital has released an update to its RedLaser app (iTunes link, $1.99) which speeds up barcode scanning by not requiring you to first take a picture. Occipital claims that its 'realtime barcode scanner' is the only one which works on phones with both autofocus (the new 3G S) and without (the older iPhone and iPod Touch). Other barcode scanning apps, such as ShopSavvy's, can also take advantage of the autofocus camera on the 3G S, but can't do on-the-fly scanning on the older models."); ScanZoom (Attia) at 7:62-8:3 ("However, if the windows message loop is empty, the user is then directed to push a button on mobile device 105 to take an image of barcode 103 in step 363. Alternatively, the image may be acquired automatically by mobile device 105 for "on-the-fly" decoding. Camera control module 305 interacts directly with API 301 to perform this step since the user must physically push a button on mobile device 105 to take the image. Alternatively, this feature may be fully automated.").

A person of ordinary skill in the art would have been motivated to incorporate these teachings into Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU

because doing so would help ensure that the image is captured when certain quality metrics are satisfied, rather than relying on the user's discretion which may not check for those quality metrics. In selecting criteria upon which to evaluate the check image, a POSITA would have been motivated to select one or more criteria, each of which would increase the likelihood that the image would be readable for use in deposit. References such as Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, and ScanZoom disclose the benefits of incorporating automatic capture based on one or more monitoring criteria into such systems. *See, e.g.*, Zhu at 7 ("…to photograph the targeted business card, the mobile phone user needs only to hold the mobile phone steady and trigger the 'shoot' button of the digital camera, either manually or automatically by the mobile phone, thus overcoming a drawback, namely the tendency of the camera to shake during shooting."); Yoon at [0007] ("In the business card recognition function, the probability of satisfactorily recognizing the business card in order to obtain the information contained in the business card varies depending on photographing conditions, such as the size, brightness, and image quality of the photographed business card. Thus, as described above, when the user photographs the business card based on his/her subjective judgment, the business card may be photographed to be too large, small or dark, thereby lowering the probability of satisfactorily recognizing the business card. Also, as described above, when the user manually selects a key button or a menu in order to photograph the business card, shaking may occur due to the selecting of the key button or the menu, and thus, shaking noise may be present in an image of the photographed business card, thereby significantly lowering the probability of satisfactorily recognizing the business card."); Luo at 5 ("The present invention ensures that the front of the object being imaged is substantially parallel to the image plane 320 of the camera system 100 to reduce the projective distortion of the image."); CAM (Pervasive Computing) at

76 (“The CamBrowser application’s primary purpose is to capture a visual image of the form and decode and interpret the embedded CamShell program, which executes the user interface that collects information from the user. CamBrowser executes instructions in sequence, starting with instruction zero. Internally, a simple virtual machine keeps track of the current program counter and local variables. You can scan codes either by pointing the camera at the CamForm (scan mode) or by explicitly taking a picture (click mode). We’ve found the former to be more suitable for quick, directed user interaction, while the latter is more effective for capturing an entire CamShell program and entering large amounts of data—for example, in a loop.”); MotoRokr (MotoRokr Manual) at 108; 98 (“Your phone automatically captures a business card image when the card is properly framed in the OCR viewfinder. (To activate or deactivate the auto-capture feature, see “business card reader settings” on page 97).”); RedLaser (TechCrunch) at 1 (“Now that the new iPhone 3G S has a video camera, TechStars startup Occipital has released an update to its RedLaser app (iTunes link, $1.99) which speeds up barcode scanning by not requiring you to first take a picture. Occipital claims that its ‘realtime barcode scanner’ is the only one which works on phones with both autofocus (the new 3G S) and without (the older iPhone and iPod Touch). Other barcode scanning apps, such as ShopSavvy’s, can also take advantage of the autofocus camera on the 3G S, but can’t do on-the-fly scanning on the older models.”); ScanZoom (Attia) at 7:62-8:3 (“However, if the windows message loop is empty, the user is then directed to push a button on mobile device 105 to take an image of barcode 103 in step 363. Alternatively, the image may be acquired automatically by mobile device 105 for “on-the-fly” decoding. Camera control module 305 interacts directly with API 301 to perform this step since the user must physically push a button on mobile device 105 to take the image. Alternatively, this feature may be fully automated.”).

A person of ordinary skill in the art would have been motivated to use MICR line detection as one such monitoring criteria. As discussed in Section V.B.1.c, numerous references teach that the MICR line is required information for check deposit, and thus teach reading the MICR line from the image. A person of ordinary skill in the art would have therefore found it obvious to use MICR line detection as an additional monitoring criterion in addition to those taught by Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, and ScanZoom. Once the other monitoring criteria are satisfied, a skilled artisan would have been motivated to further confirm that the MICR line can be detected before capturing the image. Doing so before further reading the rest of the text on the image, submitting the image, and processing the image at a remote bank server would have provided the same benefits as other monitoring criteria—for example reducing bandwidth usage and transmission delays. A person of ordinary skill in the art would have recognized that waiting to capture the image until the MICR is detected would reduce user frustration by avoiding informing the user that an image was captured only to request that the user retake the image after it is later determined the MICR line cannot be read.

It would have therefore been obvious to a person of ordinary skill in the art to automatically capture an image of a document when one or more monitoring criteria are satisfied, for example as taught in Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, and ScanZoom. Doing so would have involved simply combining prior art elements/techniques (e.g., Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, and ScanZoom's automatic capture techniques) according to known devices/methods to yield predicable results (e.g., obtaining higher quality images) and the simple substitution of one known element (e.g., other image capture techniques) for another (e.g., automatic capture techniques) to obtain predictable results (e.g., a higher quality image of a document).

A person of ordinary skill in the art would have had a reasonable expectation of success. For example, automatically capturing an image of a document would have simply involved writing computer code to evaluate the output of the mobile device camera with respect to one or more monitoring criteria and save/capture the image when that/those criteria are satisfied. Moreover, as discussed above, Luo, Zhu, Yoon, CAM, MotoRokr, RedLaser, and ScanZoom each teach that the predictable result of using an alignment guide to help capture an image of a document is an image that is less prone to problems like projective distortion and other image quality problems. The references use similar well-known components such that combining the references to automatically capture an image of a document in the proposed combination(s) would have been a routine, predictable design for a person of ordinary skill in the art and would not have involved any undue experimentation.

iii. It would have been obvious to a POSITA to provide instructions and/or feedback to the user during and after image capture ('310 and '095 Patents)

To the extent USAA contends that the prior art identified in Section V.A does not disclose providing instructions and/or feedback to the user during and after image capture, a POSITA would have found such feature obvious in view of Luo, Zhu, Yoon, ImageNet, WVUFCU, CAM, MotoRokr, RedLaser, and/or ScanZoom, at least under USAA's interpretation of the term. As an illustrative, non-limiting example, a person of ordinary skill in the art would have been motivated to combine the teachings of references such as Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (which disclose, at least in one embodiment, depositing a check using an image captured by a mobile device) with references such as Luo, Zhu, Yoon, ImageNet, WVUFCU, CAM, MotoRokr, RedLaser, and/or ScanZoom (which disclose providing instructions and/or feedback to the user before, during, and after capturing an image of a document) to assist and inform the user regarding the status of image

capture and how to take a better quality image. Such a combination would render obvious, for example, limitations containing the phrases "providing instructions to the user" and "providing feedback advising the user that the check has been captured" in claims 1, 2, 5, 6, 8, 9, 10, 11, 14, and 15 of the '310 patent and claims 1, 9, 18, and 26 of the '095 patent, and limitations containing the phrase "providing feedback to the user" in claims 1, 5, 10, and 14 of the '310 patent and in claims 2, 10, 19, and 27 of the '095 patent.

A person of ordinary skill in the art would have been motivated to combine such references because the references are all in the same field (image capture and processing) and address the same problem or objective (capturing images of sufficient quality for future processing) in similar ways (with user devices). *See e.g.*, Yoon at Abstract ("Provided are an apparatus and method for photographing a business card in a portable terminal. The method includes recognizing boundary lines of the business card from an image received from a camera when a recognition and photographing key for a business card is selected, and allowing the business card to be automatically photographed by using the recognized boundary lines."); Luo at Abstract ("A method and system for capturing the frontal image of an object having at least two substantially straight edges (205), which helps reduce projective distortion in the image."); Zhu at 7 ("An object of the present invention is to introduce a method of scanning business cards using a mobile phone with a digital camera. The method is implemented on a mobile phone having a digital camera and involves a business card scanning feature, but does not have the drawbacks of the background art."); ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page

1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone"); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that."); CAM (Parikh 2005a) at 1 ("In this paper we introduce mobile document processing as an accessible and locally harmonious way to provide information services to the developing world [2]. In our system a camera-equipped mobile phone is used for image capture and data entry. Document interaction is specified using CamShell - a narrative document-embedded programming language that can allow developers to add interactive audio, data entry, validation, processing, and networking instructions to otherwise inert paper documents."); MotoRokr (MotoRokr Manual) at 108; 98

("Your phone automatically captures a business card image when the card is properly framed in the OCR viewfinder. (To activate or deactivate the auto-capture feature, see "business card reader settings" on page 97)."), 98 ("Your phone interprets the text on the captured business card image, and sorts the text into logical categories."); RedLaser (TechCrunch) at 1 ("Now that the new iPhone 3G S has a video camera, TechStars startup Occipital has released an update to its RedLaser app (iTunes link, $1.99) which speeds up barcode scanning by not requiring you to first take a picture. Occipital claims that its 'realtime barcode scanner' is the only one which works on phones with both autofocus (the new 3G S) and without (the older iPhone and iPod Touch). Other barcode scanning apps, such as ShopSavvy's, can also take advantage of the autofocus camera on the 3G S, but can't do on-the-fly scanning on the older models."); ScanZoom (Attia) at 3:45-52 ("The user then takes a picture of the barcode using the digital camera. As soon as the barcode is captured, the ScanZoom software decodes the barcode utilizing a decoding engine integral to the ScanZoom software. Alternatively, the Scanzoom software may send the image of the barcode to a central server for decoding by sending a multimedia message service ("MMS") message to the server containing the barcode image.").

As explained in Section V.B.1.j and k, it was well known to provide instructions and/or feedback to a user before, during, and after image capture with a mobile device, including to provide help the user satisfy a monitoring criteria, to help the user properly situate the document to be captured in a preview window to produce a higher quality image of the document. *See, e.g.*, Luo at 6 ("Various techniques may be used to indicate to the user that an appropriate number of reference lines 135 are substantially parallel to the respective straight edges 205 of the object depicted in the preview window 125. For example, an alarm composed of sounds, such as a clicking sound output from the camera system 100, may be provided. In the same way, it may

be the light illuminated in the preview window 125 or the light illuminated elsewhere in the camera system 100. In addition, an image indicator may be displayed on the preview window 125, for example, when each reference line 135 is substantially parallel to a corresponding straight edge 205, the reference line 135 may change color or flicker individually. As an alternative, when the straight edge 205 displayed on the preview window 125 is substantially parallel to the corresponding reference line 135, the system 100 can automatically capture the selected image and provide the user with or without instructions."); Yoon at [0026] ("If it is determined that the current brightness is not appropriate for photographing the business card, the portable terminal displays a message indicating this fact on the display unit 106 in step 227, and then returns to operation 205."), [0028] ("Next, the portable terminal determines whether all the upper, lower, left and right reference boundary lines coincide with the boundary lines of the business card in step 215. If all the reference boundary lines do not coincide with all the boundary lines of the business card, the portable terminal returns to step 209. If all the reference boundary lines completely coincide with the boundary lines of the business card as illustrated in FIG. 3D, the portable terminal proceeds to step 217 and displays a message informing the user of the start of automatic photographing for the business card. Here, it is possible to inform the user of the start of automatic photographing by generating a predetermined alarm sound of automatic photographing or changing the shapes, thicknesses or colors of the reference boundary lines."); Zhu at 9 ("The technical solution of the present invention is further characterized in that said camera system displays on said screen instructional text guiding the mobile phone user to perform correct operations during the execution of each step."); ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to

Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone"); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that."); CAM (Parikh 2007) at 58-59 ("After data has been entered, the user can review values before submitting them to the server. By focusing the mobile phone camera on the appropriate barcode, the currently entered data is displayed on the screen, usually next to the value entered on paper (see figure 4.4). This allows both the record keeper and the SHG member to confirm that data was entered properly (mimicking the transparency of the earlier paper-based process).

By focusing on the loan purpose field, the recorded audio clip is played. If a value is incorrect, the user can capture an image of just that barcode, displaying a prompt to edit the value."); MotoRokr (MotoRokr Manual) at 108, 98 ("Your phone automatically captures a business card image when the card is properly framed in the OCR viewfinder. (To activate or deactivate the auto-capture feature, see "business card reader settings" on page 97)."); RedLaser (Review Video) at 4:49; ScanZoom (Attia) at 6:51-7:6 ("Referring next to FIG. 2A, shown is the process utilized by ScanZoom to acquire and decode barcode 103 once the application has been launched. A user first launches the ScanZoom program on mobile device 105 by selecting its icon from a list of programs located on the device in step 201. Generally, this will initialize the camera and all the network resources needed by ScanZoom for the application to function properly. Once the user has started the program, a message appears directing the user to take a picture of barcode 103 in step 203. ScanZoom provides the user with a "preview" window which functions as a viewfinder, allowing the user to properly center and align the barcode before taking a picture. Once the barcode image has been acquired by digital camera 107, ScanZoom attempts to decode the barcode utilizing its proprietary barcode decoding algorithm in step 205. If the software cannot decode barcode 103 on a first attempt in step 205, the ScanZoom software attempts to decode the image a finite amount of times. The number of decoding attempts can either be fixed or able to be altered by the user. Once barcode 103 has been properly decoded as verified in step 207, mobile device 105 prepares the decoded information for transfer to server 113 to undergo further processing in step 209.").

A person of ordinary skill in the art would have been motivated to incorporate these teachings into Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU because doing so would help ensure that user is informed of about the capture process, including

what needs to be done to obtain a high quality image and when such metrics are met. References such as Luo, Zhu, Yoon, ImageNet, WVUFCU, CAM, MotoRokr, RedLaser, and ScanZoom disclose the benefits of incorporating such instructions and feedback into such systems. *See, e.g.*, Zhu at 5 ("The present invention has the following advantages over the background art: 1. It uses screen prompts to guide the mobile phone user to take high-quality photographs of targeted business cards. The four sides of the targeted business card detected on said screen enable the mobile phone user to clearly view the current position and shape of the targeted business card, and the prompt arrows can let the mobile phone user clearly know how to move the targeted business card in order to take an acceptable photograph. Screen prompts guide any mobile phone user who is not adept at taking photographs to take high-quality photographs of targeted business cards. 2. As a result of the automatic shooting feature, the mobile phone camera photographs targeted business cards without shaking and thereby helps to improve the quality of photographs taken of targeted business cards. The above two advantages ensure that said mobile phone can take targeted business card photographs that are high-quality and do not contain any "noise," thus ensuring the recognition accuracy of said business card recognition system and increasing the usefulness and convenience of said business card recognition system."); Yoon at [0007] ("In the business card recognition function, the probability of satisfactorily recognizing the business card in order to obtain the information contained in the business card varies depending on photographing conditions, such as the size, brightness, and image quality of the photographed business card. Thus, as described above, when the user photographs the business card based on his/her subjective judgment, the business card may be photographed to be too large, small or dark, thereby lowering the probability of satisfactorily recognizing the business card. Also, as described above, when the user manually selects a key button or a menu in order to photograph

the business card, shaking may occur due to the selecting of the key button or the menu, and thus, shaking noise may be present in an image of the photographed business card, thereby significantly lowering the probability of satisfactorily recognizing the business card."); Luo at 5 ("The present invention ensures that the front of the object being imaged is substantially parallel to the image plane 320 of the camera system 100 to reduce the projective distortion of the image."); ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone"); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the

iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that."); CAM (Parikh 2007) at 58-59 ("After data has been entered, the user can review values before submitting them to the server. By focusing the mobile phone camera on the appropriate barcode, the currently entered data is displayed on the screen, usually next to the value entered on paper (see figure 4.4). This allows both the record keeper and the SHG member to confirm that data was entered properly (mimicking the transparency of the earlier paper-based process). By focusing on the loan purpose field, the recorded audio clip is played. If a value is incorrect, the user can capture an image of just that barcode, displaying a prompt to edit the value."); MotoRokr (MotoRokr Manual) at 108; 98 ("Your phone automatically captures a business card image when the card is properly framed in the OCR viewfinder. (To activate or deactivate the auto-capture feature, see "business card reader settings" on page 97)."); RedLaser (Review Video) at 3:53-5:19; ScanZoom (Attia) at 6:51-7:6 ("Referring next to FIG. 2A, shown is the process utilized by ScanZoom to acquire and decode barcode 103 once the application has been launched. A user first launches the ScanZoom program on mobile device 105 by selecting its icon from a list of programs located on the device in step 201. Generally, this will initialize the camera and all the network resources needed by ScanZoom for the application to function properly. Once the user has started the program, a message appears directing the user to take a picture of barcode 103 in step 203. ScanZoom provides the user with a "preview" window which functions as a viewfinder, allowing the user to properly center and align the barcode before taking a picture. Once the barcode image has been acquired by digital camera 107, ScanZoom attempts to decode the barcode utilizing its proprietary barcode decoding algorithm in step 205. If the software cannot decode barcode 103

on a first attempt in step 205, the ScanZoom software attempts to decode the image a finite amount of times. The number of decoding attempts can either be fixed or able to be altered by the user. Once barcode 103 has been properly decoded as verified in step 207, mobile device 105 prepares the decoded information for transfer to server 113 to undergo further processing in step 209.").

It would have therefore been obvious to a person of ordinary skill in the art to provide instructions and/or feedback to the user during and after image capture, for example as taught in Luo, Zhu, Yoon, ImageNet, WVUFCU, CAM, MotoRokr, RedLaser, and ScanZoom. Doing so would have involved simply combining prior art elements/techniques (e.g., display of instructions and/or feedback) according to known devices/methods to yield predicable results (e.g., a higher quality image of a document) and the simple substitution of one known element (e.g., one user interface) for another (e.g., a user interface that provides instructions and/or feedback) to obtain predictable results (e.g., a higher quality image of a document).

A person of ordinary skill in the art would have had a reasonable expectation of success. For example, providing instructions and/or feedback to the user would have simply involved writing computer code to evaluate the output of the mobile device camera with respect to one or more monitoring criteria and save/capture the image when that/those criteria are satisfied. Moreover, as discussed above, Luo, Zhu, Yoon, ImageNet, WVUFCU, CAM, MotoRokr, RedLaser, and ScanZoom each teach that the predictable result of using an alignment guide to help capture an image of a document is an image that is less prone to problems like projective distortion and other image quality problems. Providing instructions and/or feedback to the user during this process would have predictably facilitated this result by providing the user with more information about the capture process. The references use similar well-known components such

that combining the references to provide instructions and/or feedback before, during, and after the image capture process in the proposed combination(s) would have been a routine, predictable design for a person of ordinary skill in the art and would not have involved any undue experimentation.

iv. It would have been obvious to a POSITA to perform OCR on a check image to obtain/recognize the MICR and/or amount payable (’310 and ’095 Patents)

To the extent USAA contends that the prior art identified in Section V.A does not disclose performing OCR on a check image to obtain/recognize deposit information, including the MICR line and/or amount payable, a POSITA would have found such feature obvious in view of Garcia, Cohen, Fellerman, Viera, and/or Byrne, at least under USAA’s interpretation of the term. As an illustrative, non-limiting example, a person of ordinary skill in the art would have been motivated to combine the teachings of references such as Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (which disclose, at least in one embodiment, depositing a check using an image captured by a mobile device) with references such as Garcia, Cohen, Fellerman, Viera, and/or Byrne (which disclose using OCR to obtain deposit information) to obtain information from the check that was well known to be needed for deposit. Such a combination would render obvious, for example, limitations containing the phrases “optical character recognition is performed on the check,” “a magnetic ink character recognition (MICR) line on the check … is recognized,” and “an amount payable on the check … is validated” in claims 1, 2, 5, 6, 8, 9, 10, 11, 14, and 15 of the ’310 patent and claims 1, 9, 18, and 26 of the ’095 patent.

A person of ordinary skill in the art would have been motivated to combine such references because the references are all in the same field (image capture and processing) and address the same problem or objective (processing check images) in similar ways (using digital

equipment). *See e.g.*, Garcia at 4:11-18 ("The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations"); Fellerman at [0018] ("In accordance with one embodiment of the present invention, a method for processing finance payments from a purchaser to a merchant in a transaction is contemplated. The method may begin with receiving images of a plurality of financial instruments. Each of the financial instruments may include a monetary value and at least one identifier associated with a financial account of the purchaser. The method may then continue with evaluating an approval status for the transaction based upon the account identifier on the financial instruments. Thereafter, the method may include transferring funds to the merchant. The funds may be transferred in response to the evaluation of the approval status. Additionally, the method may include the step of deducting funds from the financial account of the purchaser to the processor account at prescheduled intervals. An amount corresponding to the monetary value of a sequential one of the financial instruments may be deducted at each interval."); Viera at Abstract ("Embodiments of the present invention include a computer-implemented method of performing a financial transaction on a network comprising storing a plurality of risk factors and associated severities in a web-enabled remote policy database, the severities defining actions taken when a specific risk factor is detected, storing a plurality of hold override policies that associate a deposit type with a hold type in the policy database, receiving deposit data including an image of a deposit check, applying the plurality of risk factors to the deposit data, and applying the plurality of hold override policies to the deposit data."), Cohen at [0025] ("The DirectFed™ system takes advantage of the new rules provided by

the Check 21 act. It allows merchants to deposit checks without having to deliver the physical check to a bank. The DirectFed™ clearing system has been approved by the Federal Reserve Bank for the delivery of checks via the Check 21 system. The checks, preferably delivered/routed via a third party system, such as the DirectFed™ clearing system 118, are directly deposited with the Federal Reserve Bank 130 (or Endpoint Exchange 132), completely bypassing the BOFD 128. There are a host of benefits and advantages provided by the system and method of the present invention, including, for example, the following: (1) merchant employees do not have to leave the office to deposit checks; (2) manual deposit tickets do not have to be created; (3) photocopies of checks do not have to be created, since the images of the checks are retained and available on the client system; (4) for large volume deposits, armored guards/vehicles are not required, since the physical checks are never transported to the bank; (5) rather than having paper checks delivered to the bank and then manually processed at the bank, the check images are electronically delivered to the Federal Reserve Bank, or Endpoint Exchange, for credit to the BOFD, resulting in greatly expedited availability of funds; (6) due to the increased clearance speed, returns are processed faster, giving merchants a greater opportunity to recover funds; (7) upon request of the merchant, checks can be automatically separated into those eligible for ACH and those which are not eligible, so that ACH eligible checks can be processed by ACH and the others via Check 21, preventing the requirement of the merchant manually separating checks before processing via ACH. With the addition of coupon processing (further described herein below), or without coupon processing (depending upon the required data), automated import into client Accounts Receivable (A/R) and other systems can be easily accommodated. This can result in an enormous saving of time, and elimination of many common keyboard errors. The BOFD does not need to have a physical location near the remote

merchant site. This allows a bank to expand the geographic coverage of its depositor. Similarly, a merchant with multiple locations can consolidate its accounts at a single bank, even for those locations that are not located near a bank branch. Banks that clear checks through other banks, or credit unions clearing through corporate credit unions, can bypass the intermediary and clear directly through the Federal Reserve Bank. This saves time and the cost of physically delivering the checks."); Byrne at Abstract ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); [0003] ("The invention relates to providing a plug-in to drive a device for securely transmitting documents in real time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to Scan checks using a personal computer and deposit the checks via the browser to a bank account."); [0033] ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to Scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); [0041] ("A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a

browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.").

As explained in Section V.B.1.c, it was well known to perform OCR on a check image to obtain/recognize deposit information in a remote deposit system to accurately determine, for example, from where, and how much funds would be drawn. *See, e.g.*, Garcia at 11:7-19 ("For its part, the institution that renders the service must have certain electronic means of recognizing, verifying and processing the information sent by the user. These means consist basically of computer equipment connected to the Internet that has an optical character recognition (OCR) system, a code authentication system and means of transmitting data to the institution's document processing system. The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems)"); Fellerman at [0042] ("The check scanner 28 is understood not to segment the captured data into various blocks described above. However, it is contemplated that the bank routing number 40 and the account identifier 42, which are inscribed as MICR data, may be read and recognized in addition to the captured raw image or raster data of the check 24. The MICR data may be appended to the image of the check 24, or may be transmitted separately. The data is then combined with additional information provided to the remote terminal 26 for transmission to a central payment processor 52, the details of which will be described more fully below."); Viera at 7:1-32 ("FIG. 4 illustrates a method of performing a transaction on a network according to one embodiment of the present invention. FIG. 4 illustrates a method of performing a deposit transaction. At step 401, a user inserts a check into an image scanner at a kiosk. At step 402, the back and front of the check are scanned and the images are stored (e.g. bitmaps). The magnetic encoding ("MICR") at the bottom of the check is also scanned. The user enters the

amount of the deposit using a touchscreen keypad at step 403. At step 404, the bitmap images and MICR obtained from step 402 are passed to an optical character recognition ("OCR") program. OCR will perform several functions. It will re-read the MICR using optical instead of magnetic techniques. It will detect the check amount, payee name, and check number. It will detect the imprinted name of the paying financial institution. It will determine if a signature is present on the front of the check. It will also determine if a signature is present in the endorsement section on the back of the check. A deposit type is determined at step 405 using the information obtained from step 404. In making this determination, risk policies and routing group information from database tables 406 and 407 are used. If deposit is rejected at step 408, the check is returned to the user. If the deposit is accepted, it will be examined for hold override policies using policy information stored in database 411. Funds availability is calculated and presented to the user at step 412. If the user rejects the deposit at step 413, then the check will be returned at step 414. If the user elects to proceed, deposit information will be printed on the check and stored in a secure internal bin at step 415. At step 416, the check images and deposit information will be posted through the shared branching network 417 to be received by the appropriate financial institution."); Cohen at [0022] ("Referring now to FIGS. 1 and 2, at the conclusion of either the scanning or data importation process step 202, in step 204 another process is triggered which performs OCR as well as parsing of the MICR Line information. In a subsequent step 206, the OCR and parsing processes return value data, including: (1) check amount; (2) bank routing number; (3) bank account number; and (4) check number. Significantly, these processes also return quantitative measurements, or values, representing the confidence level of the accuracy of the OCR results pertaining to each of the respective pieces of data. Subsequently, in step 208, a remote operator at the local merchant site 100 reviews the

images and data for each check in the batch. Significantly, the efficacy of this process is greatly enhanced by assigning color-coding in accordance with the respective confidence levels. Preferably, green, yellow and red colors are utilized to provide easy visual indication to the operator as to the confidence level attached to each data item. Subsequently, in step 210, and based upon a configuration setting in the profile that governs the particular batch, the operator is required to manually validate only those data items with a certain confidence level. Termed "Examination Rigor", this profile configuration value may be set to require that only low confidence data (i.e. marked red) need be inspected, or that both low and moderate confidence level data (i.e. marked red or yellow) must be inspected, or that all data (i.e. even items marked green) must be inspected for errors. Upon completion of the review, the batch is ready for transmission to the DirectFed™ clearing system 118, or a like third party system. For convenience, all further description herein will refer to transmission to the DirectFed™ clearing system 118. In step 212, the image and data information is packaged at the client 100 for subsequent transmission, in step 214, to the DirectFed™ clearing system 118 via the Internet utilizing an encrypted web service."); Byrne at [0174] ("An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic."); [0170] ("The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item").

A POSITA would have recognized that without providing a physical copy of the check, the only way to obtain deposit information would be to either read it from the check image or request a user to enter the information manually. A person of ordinary skill in the art would have

been motivated to incorporate Cohen, Fellerman, Viera, and/or Byrne's teachings into Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU because doing so would identify relevant and essential deposit information while reducing the possibility of user error. References such as Cohen, Fellerman, Viera, and/or Byrne disclose the benefits of incorporating OCR to obtain/recognize deposit information into such systems. *See, e.g.*, Fellerman at [0046] ("Next, in step 512 the check number 44 from the first of the plurality of checks to be scanned is entered via the keypad 54, and in step 512, the amount specified thereby is likewise entered via the keypad 54. Again, the screen 54 displays prompts corresponding to the respective data inputs. Then, the check scanner 28 is activated and the check 24 is fed for scanning in step 516. After scanning the front and back sides of the check 24, basic verifications are performed according to decision block 518 to ensure that the image data was properly captured. For example, if the check scanner 28 timed out before receiving any data, then an indicator would be activated to prompt a re-scan. Although the check number and check amount data entered into the remote terminal 26 via the keypad 54 corresponds to the scanned check images, the check images are understood to be stored independently in certain embodiments."), [0048] ("All of the entered data is then verified in a check summary confirmation step 524 and a detail confirmation step 526. In one embodiment, the total of all the checks 24 presented as keyed into the remote terminal 26 in step 514 may confirmed. However, other related aggregate or summary data relating to the checks 24 may also be confirmed. Those having ordinary skill in the art will readily devise such alternative summary data. In the detail confirmation step 526, all or part of the entered data, such as the identification number, date of birth, issuing state, phone number, the check number for each of the scanned checks, and so forth may be verified."); Viera at 7:1-32 ("FIG. 4 illustrates a method of performing a transaction on a network according to one

embodiment of the present invention. FIG. 4 illustrates a method of performing a deposit transaction. At step 401, a user inserts a check into an image scanner at a kiosk. At step 402, the back and front of the check are scanned and the images are stored (e.g. bitmaps). The magnetic encoding ("MICR") at the bottom of the check is also scanned. The user enters the amount of the deposit using a touchscreen keypad at step 403. At step 404, the bitmap images and MICR obtained from step 402 are passed to an optical character recognition ("OCR") program. OCR will perform several functions. It will re-read the MICR using optical instead of magnetic techniques. It will detect the check amount, payee name, and check number. It will detect the imprinted name of the paying financial institution. It will determine if a signature is present on the front of the check. It will also determine if a signature is present in the endorsement section on the back of the check. A deposit type is determined at step 405 using the information obtained from step 404. In making this determination, risk policies and routing group information from database tables 406 and 407 are used. If deposit is rejected at step 408, the check is returned to the user. If the deposit is accepted, it will be examined for hold override policies using policy information stored in database 411. Funds availability is calculated and presented to the user at step 412. If the user rejects the deposit at step 413, then the check will be returned at step 414. If the user elects to proceed, deposit information will be printed on the check and stored in a secure internal bin at step 415. At step 416, the check images and deposit information will be posted through the shared branching network 417 to be received by the appropriate financial institution."); Cohen at [0018] ("Significantly, the check images comprise a legally substitutable electronic version of the original physical checking document in compliance with Check 21 laws. Scanned electronic images can be further processed utilizing Optical Character Recognition (OCR) and pattern recognition software running on either a local stand alone workstation or a

local network server at the merchant's location. The scanned electronic images may be stored locally within an electronic database and image repository at the remote merchant location, which may also include a customer's office, a credit union office or another financial institution office. Software developed by the present inventors, particularly for use with the invention, runs on the merchant computer system to perform, in addition to or in combination with other processes, a process for verifying that each and every electronic check image is in conformance with Check 21 laws and is available for transmission to a Federal Reserve Bank branch, without requiring an intermediate step of transmission to a BOFD. In this manner, the system and methods of the present invention enable merchants to perform local on-site processing of batch checks, or imported batch check images, which are routed by the present system, for transmission to a Federal Reserve Bank for immediate processing."), [0043] ("Step 3: The scanner processes each check, including the printed endorsement on the back of each check, reading the magnetic MICR line, and providing an image of the front and back of the each check. This information is stored on disk and the process continues until the operator determines there are no more checks to scan, or the maximum number of checks in the batch, as defined in the profile, has been reached. At that time the operator can review each check and, optionally, can remove and rescan checks having potentially inadequate or poor quality images, or rotate checks scanned, for example, in an upside-down orientation. When the operator is satisfied, the batch is concluded, whereupon the images are stored in the local client image repository and the information about the batch and each check is recorded in the local client database. When the scanner captures an image of an item, a security hash value is simultaneously created and stored along with it. This hash value is used to ensure the integrity of the image and prevent any attempts to alter it from the instant it is captured by the scanner and stored on the hard drive of

the client PC, until it is packaged and uploaded to the Skyline Data DirectFed™ clearing system."); Byrne at [0174] ("An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic."); [0170] ("The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item").

It would have therefore been obvious to a person of ordinary skill in the art to perform OCR on a check image to obtain/recognize deposit information, including the MICR line and/or amount payable, for example as taught in Cohen, Fellerman, Viera, and/or Byrne. Doing so would have involved simply combining prior art elements/techniques (e.g., OCR of deposit information) according to known devices/methods to yield predicable results (e.g., obtaining such information).

A person of ordinary skill in the art would have had a reasonable expectation of success. For example, performing OCR on a check image to obtain/recognize deposit information, including the MICR line and/or amount payable would have simply involved writing computer code to implement well known OCR techniques—either on the mobile device or a remote server. The possibility of errors reading such information could have been minimized by using well known image correction techniques, as well as verifying the information through known techniques such as comparing the legal and courtesy amount, and/or comparing information read from the check image to information input by a user. The references use similar well-known components such that combining the references to perform OCR on a check image to obtain/recognize deposit information in the proposed combination(s) would have been a routine,

predictable design for a person of ordinary skill in the art and would not have involved any undue experimentation.

v. It would have been obvious to a POSITA to validate the amount payable (’310 and ’095 Patents)

To the extent USAA contends that the prior art identified in Section V.A does not disclose validating the amount payable, a POSITA would have found such feature obvious in view of Cohen, Fellerman, Viera, and/or Byrne, at least under USAA’s interpretation of the term. As an illustrative, non-limiting example, a person of ordinary skill in the art would have been motivated to combine the teachings of references such as Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (which disclose, at least in one embodiment, depositing a check using an image captured by a mobile device) with references such as Cohen, Fellerman, Viera, and/or Byrne (which disclose validating the amount payable) to produce more accurate information regarding the amount of funds to be transferred. Such a combination would render obvious, for example, limitations containing the phrase “an amount payable on the check … is validated” in claims 1, 2, 5, 6, 8, 9, 10, 11, 14, and 15 of the ’310 patent and claims 1, 9, 18, and 26 of the ’095 patent.

A person of ordinary skill in the art would have been motivated to combine such references because the references are all in the same field (image capture and processing) and address the same problem or objective (processing check images) in similar ways (using digital equipment). *See e.g.*, Fellerman at [0018] (“In accordance with one embodiment of the present invention, a method for processing finance payments from a purchaser to a merchant in a transaction is contemplated. The method may begin with receiving images of a plurality of financial instruments. Each of the financial instruments may include a monetary value and at least one identifier associated with a financial account of the purchaser. The method may then

continue with evaluating an approval status for the transaction based upon the account identifier on the financial instruments. Thereafter, the method may include transferring funds to the merchant. The funds may be transferred in response to the evaluation of the approval status. Additionally, the method may include the step of deducting funds from the financial account of the purchaser to the processor account at prescheduled intervals. An amount corresponding to the monetary value of a sequential one of the financial instruments may be deducted at each interval."); Viera at Abstract ("Embodiments of the present invention include a computer-implemented method of performing a financial transaction on a network comprising storing a plurality of risk factors and associated severities in a web-enabled remote policy database, the severities defining actions taken when a specific risk factor is detected, storing a plurality of hold override policies that associate a deposit type with a hold type in the policy database, receiving deposit data including an image of a deposit check, applying the plurality of risk factors to the deposit data, and applying the plurality of hold override policies to the deposit data."); Cohen at [0025] ("The DirectFed™ system takes advantage of the new rules provided by the Check 21 act. It allows merchants to deposit checks without having to deliver the physical check to a bank. The DirectFed™ clearing system has been approved by the Federal Reserve Bank for the delivery of checks via the Check 21 system. The checks, preferably delivered/routed via a third party system, such as the DirectFed™ clearing system 118, are directly deposited with the Federal Reserve Bank 130 (or Endpoint Exchange 132), completely bypassing the BOFD 128. There are a host of benefits and advantages provided by the system and method of the present invention, including, for example, the following: (1) merchant employees do not have to leave the office to deposit checks; (2) manual deposit tickets do not have to be created; (3) photocopies of checks do not have to be created, since the images of the checks are retained and available on

the client system; (4) for large volume deposits, armored guards/vehicles are not required, since the physical checks are never transported to the bank; (5) rather than having paper checks delivered to the bank and then manually processed at the bank, the check images are electronically delivered to the Federal Reserve Bank, or Endpoint Exchange, for credit to the BOFD, resulting in greatly expedited availability of funds; (6) due to the increased clearance speed, returns are processed faster, giving merchants a greater opportunity to recover funds; (7) upon request of the merchant, checks can be automatically separated into those eligible for ACH and those which are not eligible, so that ACH eligible checks can be processed by ACH and the others via Check 21, preventing the requirement of the merchant manually separating checks before processing via ACH. With the addition of coupon processing (further described herein below), or without coupon processing (depending upon the required data), automated import into client Accounts Receivable (A/R) and other systems can be easily accommodated. This can result in an enormous saving of time, and elimination of many common keyboard errors. The BOFD does not need to have a physical location near the remote merchant site. This allows a bank to expand the geographic coverage of its depositor. Similarly, a merchant with multiple locations can consolidate its accounts at a single bank, even for those locations that are not located near a bank branch. Banks that clear checks through other banks, or credit unions clearing through corporate credit unions, can bypass the intermediary and clear directly through the Federal Reserve Bank. This saves time and the cost of physically delivering the checks.”); Byrne at Abstract (“A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise’s computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-

in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.").

As explained in Section V.B.1.d, it was well known to validate the amount payable in a remote deposit system to accurately determine the amount of funds to be transferred, including by evaluating the accuracy of the information or comparing information read from the check to itself or other available information. *See, e.g.*, Fellerman at [0046] ("Next, in step 512 the check number 44 from the first of the plurality of checks to be scanned is entered via the keypad 54, and in step 512, the amount specified thereby is likewise entered via the keypad 54. Again, the screen 54 displays prompts corresponding to the respective data inputs. Then, the check scanner 28 is activated and the check 24 is fed for scanning in step 516. After scanning the front and back sides of the check 24, basic verifications are performed according to decision block 518 to ensure that the image data was properly captured. For example, if the check scanner 28 timed out before receiving any data, then an indicator would be activated to prompt a re-scan. Although the check number and check amount data entered into the remote terminal 26 via the keypad 54 corresponds to the scanned check images, the check images are understood to be stored independently in certain embodiments."), [0048] ("All of the entered data is then verified in a check summary confirmation step 524 and a detail confirmation step 526. In one embodiment, the total of all the checks 24 presented as keyed into the remote terminal 26 in step 514 may confirmed. However, other related aggregate or summary data relating to the checks 24 may also be confirmed. Those having ordinary skill in the art will readily devise such alternative summary data. In the detail confirmation step 526, all or part of the entered data, such as the identification number, date of birth, issuing state, phone number, the check number for each of the scanned checks, and so forth may be verified."); Viera at 7:1-32 ("FIG. 4 illustrates a method of

performing a transaction on a network according to one embodiment of the present invention. FIG. 4 illustrates a method of performing a deposit transaction. At step 401, a user inserts a check into an image scanner at a kiosk. At step 402, the back and front of the check are scanned and the images are stored (e.g. bitmaps). The magnetic encoding ("MICR") at the bottom of the check is also scanned. The user enters the amount of the deposit using a touchscreen keypad at step 403. At step 404, the bitmap images and MICR obtained from step 402 are passed to an optical character recognition ("OCR") program. OCR will perform several functions. It will re-read the MICR using optical instead of magnetic techniques. It will detect the check amount, payee name, and check number. It will detect the imprinted name of the paying financial institution. It will determine if a signature is present on the front of the check. It will also determine if a signature is present in the endorsement section on the back of the check. A deposit type is determined at step 405 using the information obtained from step 404. In making this determination, risk policies and routing group information from database tables 406 and 407 are used. If deposit is rejected at step 408, the check is returned to the user. If the deposit is accepted, it will be examined for hold override policies using policy information stored in database 411. Funds availability is calculated and presented to the user at step 412. If the user rejects the deposit at step 413, then the check will be returned at step 414. If the user elects to proceed, deposit information will be printed on the check and stored in a secure internal bin at step 415. At step 416, the check images and deposit information will be posted through the shared branching network 417 to be received by the appropriate financial institution."), 7:32-64 ("FIGS. 5A-E are example graphical user interfaces illustrating a deposit transaction according to another embodiment of the present invention. FIG. 5A illustrates a menu of services 501 that is presented to the user. The user will select a service by touching a labeled button 502. If the user wants to

end the transaction, he or she can touch the exit button 503. FIG. 5B illustrates a listing of the user's accounts 504. Each account is displayed with the account type 505, the balance 506, and the available funds 507. The user selects the account that will be receiving the deposit. FIG. 5C illustrates a check deposit 508. A check is inserted into an image scanner without an envelope. The account type 509 is displayed. Verification of check scan 510 is displayed. The user uses a keypad 511 to enter the deposit amount 512. An image 513 of the check is displayed to the user. FIG. 5D illustrates an appraisal of funds availability. The account type 515 is displayed. The deposit amount 516 is also displayed. The funds that are immediately available are displayed at 517A. The availability of the remaining funds is displayed at 517B. The user can reject the deposit by touching button 519 or button 518 to accept. If the deposit is accepted, the deposit information is stored in a remote database that may be accessed by the user's financial institution. For example, after the transaction has completed, FIG. 5E illustrates instant online review of the deposit. A financial institution employee can review the details of the deposit and also see an image of the actual check. Previously, a check must be placed into an envelope for deposit. The envelope then sits in a depository until a teller can process the deposit and determine the availability. A user can then find out what the funds availability is through contacting the financial institution. The institution may also send a letter to the account holder's home with the funds availability."); Cohen at [0022] ("Referring now to FIGS. 1 and 2, at the conclusion of either the scanning or data importation process step 202, in step 204 another process is triggered which performs OCR as well as parsing of the MICR Line information. In a subsequent step 206, the OCR and parsing processes return value data, including: (1) check amount; (2) bank routing number; (3) bank account number; and (4) check number. Significantly, these processes also return quantitative measurements, or values, representing the

confidence level of the accuracy of the OCR results pertaining to each of the respective pieces of data. Subsequently, in step 208, a remote operator at the local merchant site 100 reviews the images and data for each check in the batch. Significantly, the efficacy of this process is greatly enhanced by assigning color-coding in accordance with the respective confidence levels. Preferably, green, yellow and red colors are utilized to provide easy visual indication to the operator as to the confidence level attached to each data item. Subsequently, in step 210, and based upon a configuration setting in the profile that governs the particular batch, the operator is required to manually validate only those data items with a certain confidence level. Termed "Examination Rigor", this profile configuration value may be set to require that only low confidence data (i.e. marked red) need be inspected, or that both low and moderate confidence level data (i.e. marked red or yellow) must be inspected, or that all data (i.e. even items marked green) must be inspected for errors. Upon completion of the review, the batch is ready for transmission to the DirectFed™ clearing system 118, or a like third party system. For convenience, all further description herein will refer to transmission to the DirectFed™ clearing system 118. In step 212, the image and data information is packaged at the client 100 for subsequent transmission, in step 214, to the DirectFed™ clearing system 118 via the Internet utilizing an encrypted web service."); Byrne at [0058] ("An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner

device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.").

A POSITA would have recognized that without providing a physical copy of the check, the only way to obtain the deposit amount would be to either read it from the check image or request a user to enter the information manually. A person of ordinary skill in the art would have been motivated to incorporate Cohen, Fellerman, Viera, and/or Byrne's teachings into Garcia,

Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU to ensure that the correct amount of funds are deposited while reducing the possibility of user error. References such as Cohen, Fellerman, Viera, and/or Byrne disclose the benefits of incorporating amount validation into such systems. *See, e.g.*, Fellerman at [0046] ("Next, in step 512 the check number 44 from the first of the plurality of checks to be scanned is entered via the keypad 54, and in step 512, the amount specified thereby is likewise entered via the keypad 54. Again, the screen 54 displays prompts corresponding to the respective data inputs. Then, the check scanner 28 is activated and the check 24 is fed for scanning in step 516. After scanning the front and back sides of the check 24, basic verifications are performed according to decision block 518 to ensure that the image data was properly captured. For example, if the check scanner 28 timed out before receiving any data, then an indicator would be activated to prompt a re-scan. Although the check number and check amount data entered into the remote terminal 26 via the keypad 54 corresponds to the scanned check images, the check images are understood to be stored independently in certain embodiments."), [0048] ("All of the entered data is then verified in a check summary confirmation step 524 and a detail confirmation step 526. In one embodiment, the total of all the checks 24 presented as keyed into the remote terminal 26 in step 514 may confirmed. However, other related aggregate or summary data relating to the checks 24 may also be confirmed. Those having ordinary skill in the art will readily devise such alternative summary data. In the detail confirmation step 526, all or part of the entered data, such as the identification number, date of birth, issuing state, phone number, the check number for each of the scanned checks, and so forth may be verified."); Viera at 7:1-32 ("FIG. 4 illustrates a method of performing a transaction on a network according to one embodiment of the present invention. FIG. 4 illustrates a method of performing a deposit transaction. At step 401, a user inserts a check into an image scanner at a

kiosk. At step 402, the back and front of the check are scanned and the images are stored (e.g. bitmaps). The magnetic encoding ("MICR") at the bottom of the check is also scanned. The user enters the amount of the deposit using a touchscreen keypad at step 403. At step 404, the bitmap images and MICR obtained from step 402 are passed to an optical character recognition ("OCR") program. OCR will perform several functions. It will re-read the MICR using optical instead of magnetic techniques. It will detect the check amount, payee name, and check number. It will detect the imprinted name of the paying financial institution. It will determine if a signature is present on the front of the check. It will also determine if a signature is present in the endorsement section on the back of the check. A deposit type is determined at step 405 using the information obtained from step 404. In making this determination, risk policies and routing group information from database tables 406 and 407 are used. If deposit is rejected at step 408, the check is returned to the user. If the deposit is accepted, it will be examined for hold override policies using policy information stored in database 411. Funds availability is calculated and presented to the user at step 412. If the user rejects the deposit at step 413, then the check will be returned at step 414. If the user elects to proceed, deposit information will be printed on the check and stored in a secure internal bin at step 415. At step 416, the check images and deposit information will be posted through the shared branching network 417 to be received by the appropriate financial institution."), 7:32-64 ("FIGS. 5A-E are example graphical user interfaces illustrating a deposit transaction according to another embodiment of the present invention. FIG. 5A illustrates a menu of services 501 that is presented to the user. The user will select a service by touching a labeled button 502. If the user wants to end the transaction, he or she can touch the exit button 503. FIG. 5B illustrates a listing of the user's accounts 504. Each account is displayed with the account type 505, the balance 506, and the available funds 507. The user selects the

account that will be receiving the deposit. FIG. 5C illustrates a check deposit 508. A check is inserted into an image scanner without an envelope. The account type 509 is displayed. Verification of check scan 510 is displayed. The user uses a keypad 511 to enter the deposit amount 512. An image 513 of the check is displayed to the user. FIG. 5D illustrates an appraisal of funds availability. The account type 515 is displayed. The deposit amount 516 is also displayed. The funds that are immediately available are displayed at 517A. The availability of the remaining funds is displayed at 517B. The user can reject the deposit by touching button 519 or button 518 to accept. If the deposit is accepted, the deposit information is stored in a remote database that may be accessed by the user's financial institution. For example, after the transaction has completed, FIG. 5E illustrates instant online review of the deposit. A financial institution employee can review the details of the deposit and also see an image of the actual check. Previously, a check must be placed into an envelope for deposit. The envelope then sits in a depository until a teller can process the deposit and determine the availability. A user can then find out what the funds availability is through contacting the financial institution. The institution may also send a letter to the account holder's home with the funds availability."); Cohen at [0042] ("Step 3: The scanner processes each check, including the printed endorsement on the back of each check, reading the magnetic MICR line, and providing an image of the front and back of the each check. This information is stored on disk and the process continues until the operator determines there are no more checks to scan, or the maximum number of checks in the batch, as defined in the profile, has been reached. At that time the operator can review each check and, optionally, can remove and rescan checks having potentially inadequate or poor quality images, or rotate checks scanned, for example, in an upside-down orientation. When the operator is satisfied, the batch is concluded, whereupon the images are stored in the local client

image repository and the information about the batch and each check is recorded in the local client database. When the scanner captures an image of an item, a security hash value is simultaneously created and stored along with it. This hash value is used to ensure the integrity of the image and prevent any attempts to alter it from the instant it is captured by the scanner and stored on the hard drive of the client PC, until it is packaged and uploaded to the Skyline Data DirectFed™ clearing system."), [0022] ("Referring now to FIGS. 1 and 2, at the conclusion of either the scanning or data importation process step 202, in step 204 another process is triggered which performs OCR as well as parsing of the MICR Line information. In a subsequent step 206, the OCR and parsing processes return value data, including: (1) check amount; (2) bank routing number; (3) bank account number; and (4) check number. Significantly, these processes also return quantitative measurements, or values, representing the confidence level of the accuracy of the OCR results pertaining to each of the respective pieces of data. Subsequently, in step 208, a remote operator at the local merchant site 100 reviews the images and data for each check in the batch. Significantly, the efficacy of this process is greatly enhanced by assigning color-coding in accordance with the respective confidence levels. Preferably, green, yellow and red colors are utilized to provide easy visual indication to the operator as to the confidence level attached to each data item. Subsequently, in step 210, and based upon a configuration setting in the profile that governs the particular batch, the operator is required to manually validate only those data items with a certain confidence level. Termed "Examination Rigor", this profile configuration value may be set to require that only low confidence data (i.e. marked red) need be inspected, or that both low and moderate confidence level data (i.e marked red or yellow) must be inspected, or that all data (i.e. even items marked green) must be inspected for errors. Upon completion of the review, the batch is ready for transmission to the DirectFed™ clearing system 118, or a like

third party system. For convenience, all further description herein will refer to transmission to the DirectFed™ clearing system 118. In step 212, the image and data information is packaged at the client 100 for subsequent transmission, in step 214, to the DirectFed™ clearing system 118 via the Internet utilizing an encrypted web service."); Byrne at [0058] ("An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check

Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.").

It would have therefore been obvious to a person of ordinary skill in the art to validate the amount payable, for example as taught in Cohen, Fellerman, Viera, and/or Byrne. Doing so would have involved simply combining prior art elements/techniques (e.g., validating the amount payable) according to known devices/methods to yield predicable results (e.g., obtaining deposit information).

A person of ordinary skill in the art would have had a reasonable expectation of success. For example, validating the amount payable would have simply involved writing computer code to validate the amount payable, for example by providing a confidence interval for the information read from the image, comparing one indication of the amount payable (e.g., the legal amount) to another (e.g., the courtesy amount) read from the image, and/or comparing the amount payable read from the image to an amount entered by a user. Such techniques would have predictably increased the system's confidence in the amount payable. The references use similar well-known components such that combining the references to validate the amount payable in the proposed combination(s) would have been a routine, predictable design for a person of ordinary skill in the art and would not have involved any undue experimentation.

vi. <u>It would have been obvious to a POSITA to confirm that check deposit can go forward ('310 and '095 Patents)</u>

To the extent USAA contends that the prior art identified in Section V.A does not disclose confirming that check deposit can go forward, a POSITA would have found such feature obvious in view of Cohen, Fellerman, Viera, WVUFCU, and/or Byrne, at least under USAA's interpretation of the term. As an illustrative, non-limiting example, a person of ordinary skill in the art would have been motivated to combine the teachings of references such as Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (which disclose, at least in one embodiment, depositing a check using an image captured by a mobile device) with references such as Cohen, Fellerman, Viera, WVUFCU, and/or Byrne (which disclose confirming that deposit can go forward) to ensure the user does not submit erroneous information or proceed unwillingly with deposit. Such a combination would render obvious, for example, limitations containing the phrases "[before the submitting], confirming that the mobile check deposit can go forward after optical character recognition is performed on the check [in the digital image], a magnetic ink character recognition (MICR) line on the check [in the digital image] is recognized, and an amount payable on the check [in the digital image] is validated" in claims 1, 2, 5, 6, 8, 9, 10, 11, 14, and 15 of the '310 patent and 1, 9, 18, and 26 of the '095 patent.

A person of ordinary skill in the art would have been motivated to combine such references because the references are all in the same field (image capture and processing) and address the same problem or objective (remote check deposit) in similar ways (with various user interactions). *See e.g.*, Fellerman at [0045] ("For the remote terminal 26 to perform the aforementioned capture, aggregation, and data transmission operations, it is contemplated that the remote terminal 26 is programmed with executable instructions therefor that coordinates the operation of the check scanner 28, the keypad 54, the screen 56, and the communications module

58. Further details of such executable instructions, and in particular those relating to the capture and transmission of the checks 24, are illustrated in the flowchart of FIG. 5. The method begins with a step 500 of receiving an input from the keypad 54 to initiate a deferred payment transaction. Thereafter, in step 502, an identification card of the purchaser 10 such as a driver's license or other government-issued credential with a magnetic encoding strip is scanned with a magnetic card reader 62. Typical information encoded in a magnetic strip of an identification card includes an identification number, date of birth, and the issuing State, among others. In the absence of a magnetic encoding strip, it is also contemplated that a barcode, if present on the identification card, may be scanned with a bar code reader to capture the same information. If it is determined in the decision block 504 that the read was not successful or that the presented identification card did not have a magnetic strip, the method continues with manual entry steps 506, 507, and 508, where the keypad 54 is used to enter the identification number, a code corresponding to the issuing State, and the date of birth, respectively. Otherwise, such data is read directly from the identification card. Preliminary data collection concludes with a step 510 of entering the phone number of the purchaser 10 via the keypad 54. Throughout each of these data collection steps, the screen 56 displays corresponding prompts therefor to request proper data from the user/merchant 14."); Viera at 2:62-3:16 ("Financial kiosk 110 may include software and hardware components, such as a user interface 113 and devices 112 controlled by session unit 111. User interface 113 may include a display and touch screen for providing input and output interface functionality. Session unit 111 may include software components for managing user transaction flow. Device 112 may include a card reader, check scanner, check printer, and optical character recognition scanner. Request manager 114, on the client side, may include software and hardware components for sending and receiving procedures to and from the

request manager 120 on the server side. Request manager 120 may include software and hardware components for controlling request activities between kiosk 110 and server 130 through an Internet-based network 121. Server 130 may include software and hardware components to enable communication between kiosk 110, database 131, and host 140. Server 130 processes data received from kiosk 110. It may send the data to be stored in or queried against database 131. It may also send data over an Internet-based network 132 to a shared branching network 150 through a host 140. Database 131 is a web-maintained central depository of policies and configuration items."), Cohen at [0022] ("Referring now to FIGS. 1 and 2, at the conclusion of either the scanning or data importation process step 202, in step 204 another process is triggered which performs OCR as well as parsing of the MICR Line information. In a subsequent step 206, the OCR and parsing processes return value data, including: (1) check amount; (2) bank routing number; (3) bank account number; and (4) check number. Significantly, these processes also return quantitative measurements, or values, representing the confidence level of the accuracy of the OCR results pertaining to each of the respective pieces of data. Subsequently, in step 208, a remote operator at the local merchant site 100 reviews the images and data for each check in the batch. Significantly, the efficacy of this process is greatly enhanced by assigning color-coding in accordance with the respective confidence levels. Preferably, green, yellow and red colors are utilized to provide easy visual indication to the operator as to the confidence level attached to each data item. Subsequently, in step 210, and based upon a configuration setting in the profile that governs the particular batch, the operator is required to manually validate only those data items with a certain confidence level. Termed "Examination Rigor", this profile configuration value may be set to require that only low confidence data (i.e. marked red) need be inspected, or that both low and moderate confidence

level data (i.e. marked red or yellow) must be inspected, or that all data (i.e. even items marked green) must be inspected for errors. Upon completion of the review, the batch is ready for transmission to the DirectFed™ clearing system 118, or a like third party system. For convenience, all further description herein will refer to transmission to the DirectFed™ clearing system 118. In step 212, the image and data information is packaged at the client 100 for subsequent transmission, in step 214, to the DirectFed™ clearing system 118 via the Internet utilizing an encrypted web service."); Byrne at Abstract, [0050]-[0056] ("One embodiment of the invention provides a helper application that facilitates the following: scanning of checks, front and back; MICR decoding; compression and encryption; allowing user to enter check amount and annotation; single submission and batching of scanned checks; and uploading one or a batch of checks to a staging repository via database and/or FTP uploads."); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

As explained in Section V.B.1.e, it was well known to confirm that deposit can go forward in a remote check deposit system, including after reading information from the check and before or after transmitting the check to a remote server for further processing. *See, e.g.*, Fellerman at [0046] ("Next, in step 512 the check number 44 from the first of the plurality of

checks to be scanned is entered via the keypad 54, and in step 512, the amount specified thereby is likewise entered via the keypad 54. Again, the screen 54 displays prompts corresponding to the respective data inputs. Then, the check scanner 28 is activated and the check 24 is fed for scanning in step 516. After scanning the front and back sides of the check 24, basic verifications are performed according to decision block 518 to ensure that the image data was properly captured. For example, if the check scanner 28 timed out before receiving any data, then an indicator would be activated to prompt a re-scan. Although the check number and check amount data entered into the remote terminal 26 via the keypad 54 corresponds to the scanned check images, the check images are understood to be stored independently in certain embodiments."), [0048] ("All of the entered data is then verified in a check summary confirmation step 524 and a detail confirmation step 526. In one embodiment, the total of all the checks 24 presented as keyed into the remote terminal 26 in step 514 may confirmed. However, other related aggregate or summary data relating to the checks 24 may also be confirmed. Those having ordinary skill in the art will readily devise such alternative summary data. In the detail confirmation step 526, all or part of the entered data, such as the identification number, date of birth, issuing state, phone number, the check number for each of the scanned checks, and so forth may be verified."); Viera at 7:1-32 ("FIG. 4 illustrates a method of performing a transaction on a network according to one embodiment of the present invention. FIG. 4 illustrates a method of performing a deposit transaction. At step 401, a user inserts a check into an image scanner at a kiosk. At step 402, the back and front of the check are scanned and the images are stored (e.g. bitmaps). The magnetic encoding ("MICR") at the bottom of the check is also scanned. The user enters the amount of the deposit using a touchscreen keypad at step 403. At step 404, the bitmap images and MICR obtained from step 402 are passed to an optical character recognition ("OCR") program. OCR

will perform several functions. It will re-read the MICR using optical instead of magnetic techniques. It will detect the check amount, payee name, and check number. It will detect the imprinted name of the paying financial institution. It will determine if a signature is present on the front of the check. It will also determine if a signature is present in the endorsement section on the back of the check. A deposit type is determined at step 405 using the information obtained from step 404. In making this determination, risk policies and routing group information from database tables 406 and 407 are used. If deposit is rejected at step 408, the check is returned to the user. If the deposit is accepted, it will be examined for hold override policies using policy information stored in database 411. Funds availability is calculated and presented to the user at step 412. If the user rejects the deposit at step 413, then the check will be returned at step 414. If the user elects to proceed, deposit information will be printed on the check and stored in a secure internal bin at step 415. At step 416, the check images and deposit information will be posted through the shared branching network 417 to be received by the appropriate financial institution."), 7:32-64 ("FIGS. 5A-E are example graphical user interfaces illustrating a deposit transaction according to another embodiment of the present invention. FIG. 5A illustrates a menu of services 501 that is presented to the user. The user will select a service by touching a labeled button 502. If the user wants to end the transaction, he or she can touch the exit button 503. FIG. 5B illustrates a listing of the user's accounts 504. Each account is displayed with the account type 505, the balance 506, and the available funds 507. The user selects the account that will be receiving the deposit. FIG. 5C illustrates a check deposit 508. A check is inserted into an image scanner without an envelope. The account type 509 is displayed. Verification of check scan 510 is displayed. The user uses a keypad 511 to enter the deposit amount 512. An image 513 of the check is displayed to the user. FIG. 5D illustrates an appraisal of funds availability. The account

type 515 is displayed. The deposit amount 516 is also displayed. The funds that are immediately available are displayed at 517A. The availability of the remaining funds is displayed at 517B. The user can reject the deposit by touching button 519 or button 518 to accept. If the deposit is accepted, the deposit information is stored in a remote database that may be accessed by the user's financial institution. For example, after the transaction has completed, FIG. 5E illustrates instant online review of the deposit. A financial institution employee can review the details of the deposit and also see an image of the actual check. Previously, a check must be placed into an envelope for deposit. The envelope then sits in a depository until a teller can process the deposit and determine the availability. A user can then find out what the funds availability is through contacting the financial institution. The institution may also send a letter to the account holder's home with the funds availability."); Cohen at [0022] ("Referring now to FIGS. 1 and 2, at the conclusion of either the scanning or data importation process step 202, in step 204 another process is triggered which performs OCR as well as parsing of the MICR Line information. In a subsequent step 206, the OCR and parsing processes return value data, including: (1) check amount; (2) bank routing number; (3) bank account number; and (4) check number. Significantly, these processes also return quantitative measurements, or values, representing the confidence level of the accuracy of the OCR results pertaining to each of the respective pieces of data. Subsequently, in step 208, a remote operator at the local merchant site 100 reviews the images and data for each check in the batch. Significantly, the efficacy of this process is greatly enhanced by assigning color-coding in accordance with the respective confidence levels. Preferably, green, yellow and red colors are utilized to provide easy visual indication to the operator as to the confidence level attached to each data item. Subsequently, in step 210, and based upon a configuration setting in the profile that governs the particular batch, the operator is

required to manually validate only those data items with a certain confidence level. Termed "Examination Rigor", this profile configuration value may be set to require that only low confidence data (i.e. marked red) need be inspected, or that both low and moderate confidence level data (i.e. marked red or yellow) must be inspected, or that all data (i.e. even items marked green) must be inspected for errors. Upon completion of the review, the batch is ready for transmission to the DirectFed™ clearing system 118, or a like third party system. For convenience, all further description herein will refer to transmission to the DirectFed™ clearing system 118. In step 212, the image and data information is packaged at the client 100 for subsequent transmission, in step 214, to the DirectFed™ clearing system 118 via the Internet utilizing an encrypted web service."); Byrne at [0167] ("Deposit bag is a holding/receiving area until the customer commits the deposit. Items are not committed to ICING until the bag is closed;"), [0191] ("The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710."); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone … Step 2: Resize or adjust your image Step 3: Touch "iDeposit" to send your image"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the

innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that."). To the extent that USAA alleges that confirming that deposit can go forward simply involves completing various processing/verification checks, such as OCR and validation steps, Cohen, Fellerman, Viera, WVUFCU, and/or Byrne also render doing so obvious. *See supra* §§ V.B.3.b.iv, V.B.3.b.v.

A person of ordinary skill in the art would have been motivated to incorporate these teachings into Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU because doing so would, for example, ensure the information read from the check image is correct, allow for batch submission of checks, and/or provide the user with one final opportunity to determine if they wish to move forward with the financial transaction, which cannot be undone after submission. A person of ordinary skill in the art would have therefore been motivated to perform the confirming after reading information from the check—whether at the user's device or at a remote server—but before submitting the check for deposit. References such as confirming that check deposit can go forward disclose the benefits of incorporating confirming that deposit can go forward into such systems. *See, e.g.*, Fellerman at [0046] ("Next, in step 512 the check number 44 from the first of the plurality of checks to be scanned is entered via the keypad 54, and in step 512, the amount specified thereby is likewise entered via the keypad 54. Again, the screen 54 displays prompts corresponding to the respective data inputs. Then, the check scanner 28 is activated and the check 24 is fed for scanning in step 516. After scanning the front and back sides of the check 24, basic verifications are performed according to decision block 518 to ensure that the image data was properly captured. For example, if the

check scanner 28 timed out before receiving any data, then an indicator would be activated to prompt a re-scan. Although the check number and check amount data entered into the remote terminal 26 via the keypad 54 corresponds to the scanned check images, the check images are understood to be stored independently in certain embodiments."), [0048] ("All of the entered data is then verified in a check summary confirmation step 524 and a detail confirmation step 526. In one embodiment, the total of all the checks 24 presented as keyed into the remote terminal 26 in step 514 may confirmed. However, other related aggregate or summary data relating to the checks 24 may also be confirmed. Those having ordinary skill in the art will readily devise such alternative summary data. In the detail confirmation step 526, all or part of the entered data, such as the identification number, date of birth, issuing state, phone number, the check number for each of the scanned checks, and so forth may be verified."); Viera at 7:1-32 ("FIG. 4 illustrates a method of performing a transaction on a network according to one embodiment of the present invention. FIG. 4 illustrates a method of performing a deposit transaction. At step 401, a user inserts a check into an image scanner at a kiosk. At step 402, the back and front of the check are scanned and the images are stored (e.g. bitmaps). The magnetic encoding ("MICR") at the bottom of the check is also scanned. The user enters the amount of the deposit using a touchscreen keypad at step 403. At step 404, the bitmap images and MICR obtained from step 402 are passed to an optical character recognition ("OCR") program. OCR will perform several functions. It will re-read the MICR using optical instead of magnetic techniques. It will detect the check amount, payee name, and check number. It will detect the imprinted name of the paying financial institution. It will determine if a signature is present on the front of the check. It will also determine if a signature is present in the endorsement section on the back of the check. A deposit type is determined at step 405 using the information obtained

from step 404. In making this determination, risk policies and routing group information from database tables 406 and 407 are used. If deposit is rejected at step 408, the check is returned to the user. If the deposit is accepted, it will be examined for hold override policies using policy information stored in database 411. Funds availability is calculated and presented to the user at step 412. If the user rejects the deposit at step 413, then the check will be returned at step 414. If the user elects to proceed, deposit information will be printed on the check and stored in a secure internal bin at step 415. At step 416, the check images and deposit information will be posted through the shared branching network 417 to be received by the appropriate financial institution."), 7:32-64 ("FIGS. 5A-E are example graphical user interfaces illustrating a deposit transaction according to another embodiment of the present invention. FIG. 5A illustrates a menu of services 501 that is presented to the user. The user will select a service by touching a labeled button 502. If the user wants to end the transaction, he or she can touch the exit button 503. FIG. 5B illustrates a listing of the user's accounts 504. Each account is displayed with the account type 505, the balance 506, and the available funds 507. The user selects the account that will be receiving the deposit. FIG. 5C illustrates a check deposit 508. A check is inserted into an image scanner without an envelope. The account type 509 is displayed. Verification of check scan 510 is displayed. The user uses a keypad 511 to enter the deposit amount 512. An image 513 of the check is displayed to the user. FIG. 5D illustrates an appraisal of funds availability. The account type 515 is displayed. The deposit amount 516 is also displayed. The funds that are immediately available are displayed at 517A. The availability of the remaining funds is displayed at 517B. The user can reject the deposit by touching button 519 or button 518 to accept. If the deposit is accepted, the deposit information is stored in a remote database that may be accessed by the user's financial institution. For example, after the transaction has completed, FIG. 5E illustrates

instant online review of the deposit. A financial institution employee can review the details of the deposit and also see an image of the actual check. Previously, a check must be placed into an envelope for deposit. The envelope then sits in a depository until a teller can process the deposit and determine the availability. A user can then find out what the funds availability is through contacting the financial institution. The institution may also send a letter to the account holder's home with the funds availability."); Cohen at [0044] ("Step 5: Upon conclusion of the recognition engine processing, the operator can choose a batch to verify. During the verification process, the operator is shown the image of a check and the four values read from the check. Next to each value is a visual, colored indication (green, yellow, or red) representing confidence levels. Green indicates that there is a high degree of confidence in the data, yellow indicates moderate confidence in the resulting data value, and red indicates low confidence in the data value. The operator is forced to review and manually correct items based on their respective confidence level coupled with the aforementioned examination rigor profile setting. After reviewing, and correcting as necessary, each item, the operator marks the batch as being verified."), ("The scanning module allows check and coupon documents to be scanned front side and back side, and allows for a customized physical endorsement to be placed upon an item, subject to the limitations of the scanner transport device. The module allows movement between items in the "batch to view" and/or "remove items" mode as required. It also allows scanning in single item and batch modes, and allows for saving of the entire state of an opened batch for subsequent resumption of scanning, or for concluding the scanning of a particular batch. Therefore, batches that are exited without concluding can simply be reopened to allow additional items to be scanned into the batch at a later time."), [0090] ("Significantly, in a further aspect of the invention, item capture and processing may include one or more associated payment

coupons. The DirectFed™ Deposit client application is designed to process batches of items that comprise a mixture of paper checks and paper payment coupons that accompany the checks. The image capture module, or scanning module, is specially designed to scan and process the checks and coupons in an interleaved fashion in the same order or sequence that they would be obtained by physically opening a series of envelopes received from a Payor. Special logic is incorporated in the capture module to allow the interleaved stream of checks and payment coupons to be associated together correctly, without the use of separator pages or other artificial means. Logically assigning and maintaining an envelope sequence number to each item, either check or coupon, in the batch accomplishes this. Thus, the interleaved checks and coupons from multiple envelopes can be loaded together, for example, into an input hopper of a scanner. A single envelope may contain a certain maximum number items comprised of any combination of checks and coupons."), [0091] ("By logically associating check and coupon items in a logical envelope, the envelope contents may be inspected and organized by the operator after scanning has occurred. A special screen is available directly from the capture module to display thumbnail-sized images of the checks and coupons pictorially, organized into envelopes in a graphical fashion, wherein one envelope is represented on the upper half of the display, and the next envelope in sequence is shown in the lower half of the display. Items (i.e., checks and/or coupons) may be easily "moved" between the two envelopes, using a simple drag-and-drop operation. This permits an operator to easily correct input hopper loading errors, or paper jam errors, which may cause the system to erroneously associate, or "place," an item into the wrong logical envelope."); Byrne at [0167] ("Deposit bag is a holding/receiving area until the customer commits the deposit. Items are not committed to ICING until the bag is closed;"), [0050]-[0056] ("One embodiment of the invention provides a helper application that facilitates the following:

scanning of checks, front and back; MICR decoding; compression and encryption; allowing user to enter check amount and annotation; single submission and batching of scanned checks; and uploading one or a batch of checks to a staging repository via database and/or FTP uploads."); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

It would have therefore been obvious to a person of ordinary skill in the art to confirm that check deposit can go forward, for example as taught in Cohen, Fellerman, Viera, WVUFCU, and/or Byrne. Doing so would have involved simply combining prior art elements/techniques (e.g., a confirmation step) according to known devices/methods to yield predicable results (e.g., receiving final input from the user before submission).

A person of ordinary skill in the art would have had a reasonable expectation of success. For example, confirming that check deposit can go forward would have simply involved writing computer code to provide a confirmation message to the user, for example using the screen of the mobile device, and move forward with deposit after receiving confirmation from the user. The references use similar well-known components such that combining the references to confirm that deposit can go forward in the proposed combination(s) would have been a routine,

predictable design for a person of ordinary skill in the art and would not have involved any undue experimentation.

vii. It would have been obvious to a POSITA to perform mobile check deposit with a downloaded application ('310 and '095 Patents)

To the extent USAA contends that the prior art identified in Section V.A does not disclose a downloaded application, a POSITA would have found such feature obvious in view of Byrne, Meier, Ranjan, Android Developers Website, Android Website, Asai, CAM, ImageNet, WVUFCU, RedLaser, ScanZoom, or Goyal, at least under USAA's interpretation of the term. As an illustrative, non-limiting example, a person of ordinary skill in the art would have been motivated to combine the teachings of references such as Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (which disclose, at least in one embodiment, depositing a check using an image captured by a mobile device) with references such as Byrne, Meier, Ranjan, Android Developers Website, Android Website, Asai, CAM, ImageNet, WVUFCU, RedLaser, ScanZoom, or Goyal (which disclose downloaded applications) to provide a software architecture for supporting the functionality disclosed in Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (and other references described herein) and a mechanism for putting such software on the mobile device. Such a combination would render obvious, for example, limitations containing the phrase "an app for download to a mobile device of a user" in claims 1, 2, 10, 11 of the '310 patent; and limitations containing the phrases "a downloaded app" in claims 5, 6, 8, 9, 14, 15 of the '310 patent and claims 1, 9, 18, and 26 of the '095 patent.

A person of ordinary skill in the art would have been motivated to combine such references because the references are all in the same field (image capture and processing) and address the same problem or objective (mobile devices that access a camera) in similar ways (using software on a user's device). *See e.g.*, Meier at 315 ("In this chapter, you'll take a closer

look at some of these hardware APIs. In particular, you'll learn how to play and record multimedia content including audio, video, and still images, as well as use the camera to capture images and preview and capture live video"); Goyal at 174 ("The getSnapshot() method in VideoControl allows you to take a snapshot of the image that is playing on the device screen. Of course, because the getSnapshot () method is a method of the VideoControl class, you can take an image of any video that is playing on a device screen, not just captured video. The method returns the image data as a byte array, which you can then manipulate in any way you want using the LCDUI Image class. … As you may expect by now, the locator to hook into a device camera taJ<es the form of capture://video. This gives you a Player instance that returns video in the default format. The Motorola C975 doesn't support video capture, but the default (and the only) format supported by the BenQ M75 device is video/3gpp."); Ranjan at 1 ("A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary."), Asai at Abstract ("To provide a mechanism that enables preparation of a program for photographing regardless of the type of camera device and obtaining image data in a unified format. Mobile phone 10 comprises: means for selecting a camera device; means for generating, in a case that image data of said object photographed by a selected camera device is to be stored, a camera object including attributes corresponding to said camera device; means for calling a program to

photograph an object; means for photographing the object with said camera device in accordance with the program called by the camera function calling means and for storing image data representing image of said object in the camera object; and means for obtaining image data stored by the image storing means in a unified format from the camera object."); Byrne at Abstract ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); Android Developers Website (Android Market) at 1 ("With Android Market, users can easily download apps to their Android-powered phone. Users can also rate the apps they've downloaded and leave comments. These users' ratings along with anonymous usage statistics help determine how apps are ranked and presented within Android Market."), Android Developers Website (Camera) at 1 ("The Camera class is used to connect/disconnect with the camera service, set capture settings, start/stop preview, snap a picture, and retrieve frames for encoding for video."); Android Website (Open Handset) at 1 ("The Open Handset Alliance, a group of more than 30 technology and mobile companies, is developing Android: the first complete, open, and free mobile platform. To help developers get started developing new applications, we're offering an early look at the Android Software Development Kit. … Download the SDK"), Android Website (android.hardware.CameraDevice) at 1 ("android.hardware.CameraDevice The CameraDevice class is used to open/release the camera driver, set image parameters and capture a preview frame (320*240 pixels) or a full-resolution image (1280*1024 pixels)."); CAM (Parikh 2005a) at 1 ("In this paper we introduce mobile document processing as an accessible and locally harmonious way to provide information

services to the developing world [2]. In our system a camera-equipped mobile phone is used for image capture and data entry. Document interaction is specified using CamShell - a narrative document-embedded programming language that can allow developers to add interactive audio, data entry, validation, processing, and networking instructions to otherwise inert paper documents."); ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone"); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the

whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that."); RedLaser (TechCrunch) at 1 ("Now that the new iPhone 3G S has a video camera, TechStars startup Occipital has released an update to its RedLaser app (iTunes link, $1.99) which speeds up barcode scanning by not requiring you to first take a picture. Occipital claims that its 'realtime barcode scanner' is the only one which works on phones with both autofocus (the new 3G S) and without (the older iPhone and iPod Touch). Other barcode scanning apps, such as ShopSavvy's, can also take advantage of the autofocus camera on the 3G S, but can't do on-the-fly scanning on the older models."); ScanZoom (Attia) at 3:45-52 ("3:45-52 ("The user then takes a picture of the barcode using the digital camera. As soon as the barcode is captured, the ScanZoom software decodes the barcode utilizing a decoding engine integral to the ScanZoom software. Alternatively, the Scanzoom software may send the image of the barcode to a central server for decoding by sending a multimedia message service ("MMS") message to the server containing the barcode image.").

As explained in Section V.B.1.f, it was well known to perform mobile check deposit using a downloaded application. *See, e.g.*, Meier at 4-5 ("Android phones will normally come with a suite of preinstalled applications including, but not limited to: … The Android Marketplace client for downloading thied-party [*sic*] Android applications."); Goyal at 66 ("Of course, simply requesting permissions in the JAD file is only half the story. Any rogue MIDlet could request the permission this way and cause trouble. You still need to establish some sort of trust principle between the MIDlet and the AMS; the AMS will then trust your MIDlet enough to let it be installed in a trusted domain. This is done by signing the MIDlet with a certificate that can be validated up to a root certificate installed on the device.… After the MIDlet is signed, you can test it by using the Run via OTA menu option shown in Figure 5-6. You don't use the Run

menu option like you have been doing so far because when you run the MIDlet locally (by selecting the Run menu option), it does not get installed into a protection domain. Only when installed via the Run via OTA option does your MIDlet gets a chance to be installed into a requested domain, which is the domain that you request by specifying permission for it in the JAD file."); Ranjan at 2 ("The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too."); Asai at [0025] ("JAM downloads JAR file and ADF (Application Descriptor File) to mobile phone 10. JAR file is stored in JAR storage 110 a. Written in ADF is information related to JAR file, such as AppClass key specifying the main class name used when activating the corresponding Java-AP or Spsize indicating the capacity of scratch pad 110 b allocated to the Java-AP. JAM refers to the descriptive content of the ADF and manages the execution and the behavior of Java-AP corresponding to the ADF. The management in executing Java-AP involves activation and shut down of Java-AP. The management in behavior of Java-AP includes the bridging of Java-AP and native application program and access control to data stored in non-volatile memory 110."), [0076] ("Referring now to FIG. 3, the operation example of CPU 105 will be explained, wherein CPU executes a Java-AP (hereinafter, "camera AP") by compiling a program that is written a method defined by above described camera API and above explained application API as an executable byte code JAR file including

a camera AP and ADF corresponding to the file are assumed to be stored in non-volatile memory 110 after being downloaded to mobile phone 10 from a server device."), [0138] ("(7) In the above embodiment, software having function of camera API is preinstalled in storage unit 107 of mobile phone 10. Software with such a function may also be downloaded from a mobile communications network, or storage medium storing such software may be applied to mobile phone 10."); Byrne at Abstract ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."), [0003] ("The invention relates to providing a plug-in to drive a device for securely transmitting documents in real time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to Scan checks using a personal computer and deposit the checks via the browser to a bank account."), [0033] ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to Scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."), [0041] ("A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and

deposit the checks via the browser to a bank account is provided."), [0045]-[0049], FIGs. 1-7; Android Developers Website (Android Market) at 1 ("With Android Market, users can easily download apps to their Android-powered phone. Users can also rate the apps they've downloaded and leave comments. These users' ratings along with anonymous usage statistics help determine how apps are ranked and presented within Android Market."); Android Website (Open Handset) at 1 ("The Open Handset Alliance, a group of more than 30 technology and mobile companies, is developing Android: the first complete, open, and free mobile platform. To help developers get started developing new applications, we're offering an early look at the Android Software Development Kit. … Download the SDK"); CAM (Parikh 2007) at 52 ("CAM provides an API for accessing the mobile phone's user interface, networking and multimedia input and output capabilities. Applications are downloaded on demand from an online source, either via the web or a multimedia message (MMS). The XML is cached locally on the phone along with any associated files, in a directory structure organized according to the data server and application ID."); ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit

and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone"); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that."); RedLaser (Review Video) at 1:06-3:04, RedLaser (SDK) at 1; ScanZoom (Attia) at 3:34-43 ("To utilize the ScanZoom software, a user downloads the ScanZoom onto his/her cell phone or PDA through wireless access protocol ("WAP"), infrared, or Bluetooth® connectivity. However, any protocol which allows a user to download a program to a mobile device may be utilized to download ScanZoom. Once the user has downloaded ScanZoom, the user launches the application. This causes ScanZoom to properly initialize the digital camera coupled to the mobile device to accept input. It starts the digital camera by calling its Application identification.").

A person of ordinary skill in the art would have been motivated to incorporate these teachings into Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU because doing so would, for example, provide a software architecture whereby third parties like banks could provide remote check deposit software to customers without having to preload software to mobile devices before the devices are sold or provide physical copies of storage

devices (e.g., compact disks) containing such software to users. A person of ordinary skill in the art would have recognized that downloaded applications would have therefore been a flexible and efficient way to provide software to customers to facilitate remote check deposit. References such as Byrne, Meier, Ranjan, Android Developers Website, Android Website, Asai, CAM, ImageNet, WVUFCU, RedLaser, ScanZoom, or Goyal disclose the benefits of incorporating a downloaded application into such systems. *See, e.g.*, Meier at 4-5 ("What really makes Android compelling is its open philosophy, which ensures that any deficiencies in user interface or native application design can be fixed by writing an extension or replacement. Android provides you, as a developer, the opportunity to create mobile phone interfaces and applications designed to look, feel, and function exactly as you image them. … As an application-neutral platform, Android gives you the opportunity to create applications that are as much a part of the phone as anything provided out of the box."), 11 ("All Applications Are Created Equal Android doesn't differentiate between native applications and those developed by third parties. This gives consumers unprecedented power to change the look and feel of their devices by letting them completely replace every native application with a third-party alternative that has access to the same underlying data and hardware. Every rule needs an exception and this one has two. The 'unlock' and 'in-call experience' screens can not be replaced in the initial SDK release. … Android gives developers a great opportunity. All Android applications are a native part of the phone, not just software that's run in a sandbox on top of it. Rather than writing small-screen versions of software that can be run on low-power devices, you can now write mobile applications that change the way people use their phones."), 14 ("All applications, both native and third party, are built on the application layer using the same API libraries. The application layer runs within the Android run time using the classes and services made available

from the application framework."), 142 ("Android devices are more than a simple platform for running a browser; they include location-based services, camera hardware, and accelerometers. By creating a native application, you can combine the data available online with the hardware features available on the device to provide a richer user experience"); Goyal at 1 ("MMAPI allows you to create applications for Java-enabled devices that can discover and use the multimedia capabilities of the device that they are running on. You can play different formats of audio and video files from the network, a record store, or a Java Archive (JAR) file; have advanced control over the playback of these files; capture audio and video and take snapshots; play MIDI files; generate and play back tones; stream radio over the network; and do a whole lot more."), 127 ("By being an open-ended API, MMAPI provides a single, simple yet powerful medium to embed audio and video in your MIDlets. Device manufacturers decide which part of the API they want to and can support and consequently provide implementations that take full advantage of the device's features. For example, if the device contains a camera, the MMAPI implementation on it may provide means to capture snapshots or full videos. The point is that the device manufacture is in full control of the capabilities that it wants enabled and can do so with a single API interface. Whether audio capture or live radio streaming, video capture to video playback, synchronized audio and video, or simple snapshots, MMAPI provides the means using your device. The trick is to find a device that supports all these options so you can create your magical MID let. This chapter shows you all the capabilities provided by the MMAPI to support these features."); Ranjan at 1 ("A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's

DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary."); Asai at [0025] ("JAM downloads JAR file and ADF (Application Descriptor File) to mobile phone 10. JAR file is stored in JAR storage 110 a. Written in ADF is information related to JAR file, such as AppClass key specifying the main class name used when activating the corresponding Java-AP or Spsize indicating the capacity of scratch pad 110 b allocated to the Java-AP. JAM refers to the descriptive content of the ADF and manages the execution and the behavior of Java-AP corresponding to the ADF. The management in executing Java-AP involves activation and shut down of Java-AP. The management in behavior of Java-AP includes the bridging of Java-AP and native application program and access control to data stored in non-volatile memory 110."); Byrne at [0013] ("It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits."), [0046] ("The second form is distributed with each invocation of the application software through the ubiquitous browser interface available on almost every modern personal computer. The software industry refers to this type of application as a thin client. No software is installed on the disk. The application software is the same version for all users because the plug-in is downloaded with each invocation of the application software, i.e. automatic maintenance. The plug-in does not automatically have the same level of access to

the resources that the user has. This security feature allows the user to give permission to the resource being requested, e.g. scanner, not all of the resources available to the user. The combined benefits of security with ease and speed of software maintenance makes the thin-client approach a more beneficial approach to both the customer and the enterprise."); Android Developers Website (Android Market) at 1 ("With Android Market, users can easily download apps to their Android-powered phone. Users can also rate the apps they've downloaded and leave comments. These users' ratings along with anonymous usage statistics help determine how apps are ranked and presented within Android Market. … Starting in early Q1, developers will also be able to distribute paid apps in addition to free apps. Developers will get 70% of the revenue from each purchase; the remaining amount goes to carriers and billing settlement fees—Google does not take a percentage. We believe this revenue model creates a fair and positive experience for users, developers, and carriers."); Android Website (Open Handset) at 1 ("The Open Handset Alliance, a group of more than 30 technology and mobile companies, is developing Android: the first complete, open, and free mobile platform. To help developers get started developing new applications, we're offering an early look at the Android Software Development Kit. … Download the SDK"); CAM (Parikh 2007) at 50 ("Developing custom applications for mobile phones is becoming more common. However, this requires knowledge of new APIs (some of which require licensing). Distributing applications is cumbersome. Either providers must push content to users, or users must navigate to web sites and download the software. Due to the difficulty of using the mobile web and the latency in the mobile data network, the easiest way to install an application is by using a PC to download the installer and transfering it to the phone using bluetooth or USB."); ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture

Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone"); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that."); RedLaser (SDK) at 1 ("Download the SDK (Free) This includes the scanning control that you can customize according to your application. You just need to keep the discrete Occipital logo somewhere on the scan screen."); ScanZoom (Attia) at 3:34-43 ("To utilize the ScanZoom software, a user downloads the ScanZoom onto his/her cell phone or PDA through wireless

access protocol ("WAP"), infrared, or Bluetooth® connectivity. However, any protocol which allows a user to download a program to a mobile device may be utilized to download ScanZoom. Once the user has downloaded ScanZoom, the user launches the application. This causes ScanZoom to properly initialize the digital camera coupled to the mobile device to accept input. It starts the digital camera by calling its Application identification."). In the combination, the app would be (1) associated with a bank (i.e., either the user's bank vis-à-vis providing the app itself or the bank from which funds will be drawn) and (2) control the camera (*see* § V.B.3.b.vii).

It would have therefore been obvious to a person of ordinary skill in the art to perform mobile check deposit using a downloaded application, for example as taught in Byrne, Meier, Ranjan, Android Developers Website, Android Website, Asai, CAM, ImageNet, WVUFCU, RedLaser, ScanZoom, or Goyal. Doing so would have involved simply combining prior art elements/techniques (e.g., downloaded applications) according to known devices/methods to yield predicable results (e.g., providing software that performs remote check deposit) and the simple substitution of one known element (e.g., one way to load software onto a mobile device) for another (e.g., downloading such software) to obtain predictable results (e.g., providing software that performs remote check deposit).

A person of ordinary skill in the art would have had a reasonable expectation of success. For example, performing remote check deposit on a mobile device using a downloaded app would have simply involved writing computer code using well known mobile device software development kits, such as those offered by Android, Nokia, Symbian, Java, Apple (iOS), and/or Windows. A person of ordinary skill in the art would have been aware that application stores for Android, Nokia, Apple, and Windows platforms had already been released or were in development with third party app developers, and would have recognized that downloading a

mobile check deposit application from one of such app stores or a web server or website would have been an efficient way to distribute software. The references use similar well-known components such that combining the references to perform check deposit with a downloaded application in the proposed combination(s) would have been a routine, predictable design for a person of ordinary skill in the art and would not have involved any undue experimentation.

viii. It would have been obvious to a POSITA to control the image capture process using a software abstraction layer ('310 and '095 Patents)

To the extent USAA contends that the prior art identified in Section V.A does not disclose controlling the image capture process using a software abstraction layer, such as an application programming interface (API),[16] that provides video frames (e.g., for a preview window and for capture), a POSITA would have found such feature obvious in view of Byrne, Meier, Ranjan, Android Developers Website, Android Website, Asai, CAM, ImageNet, WVUFCU, RedLaser, ScanZoom, or Goyal, at least under USAA's interpretation of the term. As an illustrative, non-limiting example, a person of ordinary skill in the art would have been motivated to combine the teachings of references such as Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (which disclose, at least in one embodiment, depositing a check using an image captured by a mobile device) with references such as Byrne, Meier, Ranjan, Android Developers Website, Android Website, Asai, CAM, ImageNet, WVUFCU, RedLaser, ScanZoom, or Goyal (which disclose controlling image capture using a software abstraction layer) to provide a software architecture for supporting the functionality disclosed in Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (and other references described herein). Such a combination would render obvious, for example,

[16] The patents provide an API as the sole example of a "software abstraction layer." *See, e.g.*, '310 patent, 11:26-28.

limitations containing the phrase "software abstraction layer" in claims and in claims 1, 2, 10, 11, 14, 15 of the '310 patent and claims 1, 18, 26 of the '095 patent; and limitations containing the phrases "video frame" and live video" in claims 1, 2, 5, 6, 8, 9, 10, 11, 14, 15 of the '310 patent and claims 1, 9, 18, and 26 of the '095 patent.

A person of ordinary skill in the art would have been motivated to combine such references because the references are all in the same field (image capture and processing) and address the same problem or objective (how to control a camera on a mobile device) in similar ways (using software on the user's device). *See e.g.*, Meier at 315 ("In this chapter, you'll take a closer look at some of these hardware APIs. In particular, you'll learn how to play and record multimedia content including audio, video, and still images, as well as use the camera to capture images and preview and capture live video"); Goyal at 174 ("The getSnapshot() method in VideoControl allows you to take a snapshot of the image that is playing on the device screen. Of course, because the getSnapshot () method is a method of the VideoControl class, you can take an image of any video that is playing on a device screen, not just captured video. The method returns the image data as a byte array, which you can then manipulate in any way you want using the LCDUI Image class. … As you may expect by now, the locator to hook into a device camera takes the form of capture://video. This gives you a Player instance that returns video in the default format. The Motorola C975 doesn't support video capture, but the default (and the only) format supported by the BenQ M75 device is video/3gpp."); Ranjan at 1 ("A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article

shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary."), Fig. 1; Asai at Abstract ("To provide a mechanism that enables preparation of a program for photographing regardless of the type of camera device and obtaining image data in a unified format. Mobile phone 10 comprises: means for selecting a camera device; means for generating, in a case that image data of said object photographed by a selected camera device is to be stored, a camera object including attributes corresponding to said camera device; means for calling a program to photograph an object; means for photographing the object with said camera device in accordance with the program called by the camera function calling means and for storing image data representing image of said object in the camera object; and means for obtaining image data stored by the image storing means in a unified format from the camera object."); Byrne at Abstract ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); Android Developers Website (Camera) at 1 ("The Camera class is used to connect/disconnect with the camera service, set capture settings, start/stop preview, snap a picture, and retrieve frames for encoding for video."); Android Website (android.hardware.CameraDevice) at 1 ("android.hardware.CameraDevice The CameraDevice class is used to open/release the camera driver, set image parameters and capture a preview frame (320*240 pixels) or a full-resolution image (1280*1024 pixels). This class does not have any public constructors. However, calling the static method open() will return a

CameraDevice object. To get a frame/picture, follow these steps: open() the driver. Open will return null if it fails. Set the image parameters (using setCaptureParams(CameraDevice.CaptureParams)). Look at CameraDevice.CaptureParams class for details. Capture (capture(Canvas)) an image with a Canvas object. Note that the Bitmap in the Canvas object must already have enough space allocated for the frame/image. If you are shooting preview, then you would want to call capture(Canvas) several times. It's not necessary to set parameters after each capture(Canvas) if the parameters are the same. Finally, close() the driver."); CAM (Parikh 2005a) at 1 ("In this paper we introduce mobile document processing as an accessible and locally harmonious way to provide information services to the developing world [2]. In our system a camera-equipped mobile phone is used for image capture and data entry. Document interaction is specified using CamShell - a narrative document-embedded programming language that can allow developers to add interactive audio, data entry, validation, processing, and networking instructions to otherwise inert paper documents."); ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be

your wallet - 9to5Mac at Page 1 (“We’ve mused on this before, but things have taken another step forward with news of Mitek’s Mobile Deposit (read, banking) application for the iPhone”); WVUFCU (iPhoneApp) at 1 (“iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone”); Finovate Blog at 2 (“But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner.”); An Eye for Technology at 1 (“With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.”); RedLaser (TechCrunch) at 1 (“Now that the new iPhone 3G S has a video camera, TechStars startup Occipital has released an update to its RedLaser app (iTunes link, $1.99) which speeds up barcode scanning by not requiring you to first take a picture. Occipital claims that its ‘realtime barcode scanner’ is the only one which works on phones with both autofocus (the new 3G S) and without (the older iPhone and iPod Touch). Other barcode scanning apps, such as ShopSavvy’s, can also take advantage of the autofocus camera on the 3G S, but can’t do on-the-fly scanning on the older models.”); ScanZoom (Attia) at 3:45-52 (“3:45-52 (“The user then takes a picture of the barcode using the digital camera. As soon as the barcode is captured, the ScanZoom software decodes the barcode utilizing a decoding engine integral to the ScanZoom software. Alternatively, the Scanzoom software may send the image of the barcode to a central server for decoding by sending a multimedia message service (“MMS”) message to the server containing the barcode image.”).

As explained in Section V.B.1.g, it was well known to use a software abstraction layer, such as an API, to control a mobile device camera, which provides live video/video frames in a mobile device to allow the user to interact with the camera, for example for displaying a preview window, recording/monitoring video, and/or capturing an image. *See, e.g.*, Meier at 315 ("In this chapter, you'll take a closer look at some of these hardware APIs. In particular, you'll learn how to play and record multimedia content including audio, video, and still images, as well as use the camera to capture images and preview and capture live video"); Goyal at 174 ("The getSnapshot() method in VideoControl allows you to take a snapshot of the image that is playing on the device screen. Of course, because the getSnapshot () method is a method of the VideoControl class, you can take an image of any video that is playing on a device screen, not just captured video. The method returns the image data as a byte array, which you can then manipulate in any way you want using the LCDUI Image class. … As you may expect by now, the locator to hook into a device camera takes the form of capture://video. This gives you a Player instance that returns video in the default format. The Motorola C975 doesn't support video capture, but the default (and the only) format supported by the BenQ M75 device is video/3gpp."); Ranjan at 1 ("A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary."); Asai at [0027] ("Application API includes a camera API. When

executing a Java-AP, KVM calls native application with a camera function through a camera API and photographs still image or moving image by camera controlling unit 104."), [0075]-[0076] ("Referring now to FIG. 3, the operation example of CPU 105 will be explained, wherein CPU executes a Java-AP (hereinafter, "camera AP") by compiling a program that is written a method defined by above described camera API and above explained application API as an executable byte code JAR file including a camera AP and ADF corresponding to the file are assumed to be stored in non-volatile memory 110 after being downloaded to mobile phone 10 from a server device."), [0017] ("As shown, a camera controlling unit 104 acts as a typical camera that photographs an object. Camera controlling unit 104 includes two types of camera devices; one camera device is for photographing still images; the other camera device is for photographing moving images. Each camera device has a lens, an image pickup device including a CCD (Charge Coupled device), an A/D converter, an image processing circuit, and a shutter. An image of an object is converted into an optical image and formed on an image pickup device. The imaging device converts the intensity of light corresponding to the optical image into quantities of electrical charge. An analogue video signal outputted from an image pickup device is converted by an A/D converter into image data in a digital video signal and displayed on a liquid crystal display 106. Image data photographed by pressing a shutter is stored temporarily in RAM 109. The image processing circuit performs compressing/expanding of image data in accordance with JPEG (Joint Photographic Experts Group image compression technology)."); Android Developers Website (Camera) at 1 ("The Camera class is used to connect/disconnect with the camera service, set capture settings, start/stop preview, snap a picture, and retrieve frames for encoding for video."), 2 ("setPreviewCallback(Camera.PreviewCallback cb) Can be called at any time to instruct the camera to use a callback for each preview frame in addition to

displaying it. … takePicture(Camera.ShutterCallbackshutter, Camera.PictureCallback raw, Camera.PictureCallback jpeg) Registers a callback to be invoked when a picture is taken.”); Android Website (android.hardware.CameraDevice) at 1 (“android.hardware.CameraDevice The CameraDevice class is used to open/release the camera driver, set image parameters and capture a preview frame (320*240 pixels) or a full-resolution image (1280*1024 pixels). This class does not have any public constructors. However, calling the static method open() will return a CameraDevice object. To get a frame/picture, follow these steps: open() the driver. Open will return null if it fails. Set the image parameters (using setCaptureParams(CameraDevice.CaptureParams)). Look at CameraDevice.CaptureParams class for details. Capture (capture(Canvas)) an image with a Canvas object. Note that the Bitmap in the Canvas object must already have enough space allocated for the frame/image. If you are shooting preview, then you would want to call capture(Canvas) several times. It’s not necessary to set parameters after each capture(Canvas) if the parameters are the same. Finally, close() the driver.”); Byrne at [0048] (“The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched.

Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114."); CAM (Parikh 2005b) at 2 ("CamShell provides an API for accessing the mobile phone's functionality - including the phone's user interface, networking and telephony features. Listed below are the main functions currently included in this API: … • Applications - These instructions are used to launch other applications, such as the Web or WAP browser."); ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone"); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other

financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that."); RedLaser (Review Video) at 3:53-5:19; ScanZoom (Attia) at 7:62-8:3 ("However, if the windows message loop is empty, the user is then directed to push a button on mobile device 105 to take an image of barcode 103 in step 363. Alternatively, the image may be acquired automatically by mobile device 105 for 'on-the-fly' decoding. Camera control module 305 interacts directly with API 301 to perform this step since the user must physically push a button on mobile device 105 to take the image. Alternatively, this feature may be fully automated.").

A person of ordinary skill in the art would have been motivated to incorporate these teachings into Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU because doing so would allow third party software developers, such as a bank, to access a mobile devices camera to facilitate mobile remote check deposit without requiring the third party to rewrite the software from the ground up. Instead, the third party would simply access the camera using APIs, which often work the same across a range of devices. Moreover, the third party would not have to update their own version of software whenever the camera requires an update—instead, the device manufacturer could update camera software without changing how the camera operates through the API software interface. References such as Byrne, Meier, Ranjan, Android Developers Website, Android Website, Asai, CAM, ImageNet, WVUFCU, RedLaser, ScanZoom, or Goyal disclose the benefits of incorporating a software abstraction

layer for accessing a camera into such systems. *See, e.g.*, Meier at 1 ("In Android, native and third-party applications are written using the same APIs and executed on the same run time. These APIs feature hardware access, location-based services, support for background services, map-based activities, relational databases, interdevice peer-to-peer messaging, and 2D and 3D graphics"), 5 ("The true appeal of Android as a development environment lies in the APIs it provides. As an application-neutral platform, Android gives you the opportunity to create applications that are as much a part of the phone as anything provided out of the box. The following list highlights some of the most noteworthy Android features: … Full multimedia hardware control including playback and recording using the camera and microphone …."), 6 ("Android includes API libraries to simplify development involving the device hardware. These ensure that you don't need to create specific implementations of your software for different devices, so you can create Android applications that work as expected on any device that supports the Android software stack. The Android SDK includes APIs for location-based hardware (such as GPS), camera, network connections, Wi-Fi, Bluetooth, accelerometers, touch screen, and power management. You can explore the possibilities of some of Android's hardware APIs in more detail in Chapter 10."), 12 ("The Android APIs The core of the SDK is the Android API libraries that provide developer access to the Android stack. These are the same libraries used at Google to create native Android applications."); Goyal at 1 ("MMAPI allows you to create applications for Java-enabled devices that can discover and use the multimedia capabilities of the device that they are running on. You can play different formats of audio and video files from the network, a record store, or a Java Archive (JAR) file; have advanced control over the playback of these files; capture audio and video and take snapshots; play MIDI files; generate and play back tones; stream radio over the network; and do a whole lot more."), 127

("By being an open-ended API, MMAPI provides a single, simple yet powerful medium to embed audio and video in your MID lets. Device manufacturers decide which part of the API they want to and can support and consequently provide implementations that take full advantage of the device's features. For example, if the device contains a camera, the MMAPI implementation on it may provide means to capture snapshots or full videos. The point is that the device manufacture is in full control of the capabilities that it wants enabled and can do so with a single API interface. Whether audio capture or live radio streaming, video capture to video playback, synchronized audio and video, or simple snapshots, MMAPI provides the means using your device. The trick is to find a device that supports all these options so you can create your magical MID let. This chapter shows you all the capabilities provided by the MMAPI to support these features."); Ranjan at 1 ("A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary."); Asai at [0024] ("KVM provides a Java-AP with API for accessing a resource of mobile phone 10. Such API includes an API for employing a class library conforming to Connected Limited Device Configuration (CLDC) and an application API for employing a class library not conforming to CLDC. By using a resource through application API, KVM is able to perform processes such as collaboration between Java-AP and a native application program, infrared communications, network connection in accordance with

Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Security (HTTPS), provision of user interface, and data saving in scratch pad 10. Unlike other native application programs, KVM is activated when a user inputs an instruction to activate it."); Android Developers Website (Camera) at 1 ("The Camera class is used to connect/disconnect with the camera service, set capture settings, start/stop preview, snap a picture, and retrieve frames for encoding for video."); Byrne at [0048] ("The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114."); CAM (Parikh 2005b) at 2 ("CamShell provides an API for accessing the mobile phone's functionality - including the phone's user interface, networking and telephony features. Listed below are the main functions currently included in this API: … • Applications - These instructions are used to launch other applications,

such as the Web or WAP browser."); ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone"); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that."); RedLaser (TechCrunch) at 1 ("Now that the new iPhone 3G S has a video camera, TechStars startup Occipital has released an update to its RedLaser app (iTunes link, $1.99) which speeds up

barcode scanning by not requiring you to first take a picture. Occipital claims that its 'realtime barcode scanner' is the only one which works on phones with both autofocus (the new 3G S) and without (the older iPhone and iPod Touch). Other barcode scanning apps, such as ShopSavvy's, can also take advantage of the autofocus camera on the 3G S, but can't do on-the-fly scanning on the older models."); ScanZoom (Attia) at 7:62-8:3 ("However, if the windows message loop is empty, the user is then directed to push a button on mobile device 105 to take an image of barcode 103 in step 363. Alternatively, the image may be acquired automatically by mobile device 105 for 'on-the-fly' decoding. Camera control module 305 interacts directly with API 301 to perform this step since the user must physically push a button on mobile device 105 to take the image. Alternatively, this feature may be fully automated.").

It would have therefore been obvious to a person of ordinary skill in the art to control the image capture process using a software abstraction layer, such as an application programming interface (API), for example as taught in Byrne, Meier, Ranjan, Android Developers Website, Android Website, Asai, CAM, ImageNet, WVUFCU, RedLaser, ScanZoom, or Goyal. Doing so would have involved simply combining prior art elements/techniques (e.g., APIs) according to known devices/methods to yield predicable results (e.g., controlling a device camera) and the simple substitution of one known element (e.g., one type of software) for another (e.g., an API) to obtain predictable results (e.g., controlling the device camera).

A person of ordinary skill in the art would have had a reasonable expectation of success. For example, controlling the image capture process using a software abstraction layer, such as an application programming interface (API) would have simply involved writing computer code that accesses a camera using well-known APIs. APIs had been well known in a number of contexts, including mobile device software development, such that a person of ordinary skill in

the art would have been familiar with them and how to write code that takes advantage of features exposed by APIs. Indeed, the existence of a plethora of APIs for controlling mobile device cameras demonstrates that implementing such APIs in mobile device applications was well within the level of ordinary skill. The references use similar well-known components such that combining the references to control the image capture process using a software abstraction layer, such as an application programming interface (API)in the proposed combination(s) would have been a routine, predictable design for a person of ordinary skill in the art and would not have involved any undue experimentation.

ix. It would have been obvious to a POSITA to receive an indication of a result of processing (’310 and ’095 Patents)

To the extent USAA contends that the prior art identified in Section V.A does not disclose receiving an indication of a result of processing, a POSITA would have found such feature obvious in view of Cohen, Fellerman, Viera, Byrne, ImageNet, and/or WVUFCU, at least under USAA’s interpretation of the term. As an illustrative, non-limiting example, a person of ordinary skill in the art would have been motivated to combine the teachings of references such as Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (which disclose, at least in one embodiment, depositing a check using an image captured by a mobile device) with references such as Cohen, Fellerman, Viera, Byrne, ImageNet, and/or WVUFCU (which disclose receiving an indication of a result of processing) to produce inform the user of the status of check processing. Such a combination would render obvious, for example, limitations containing the phrase “the app, when downloaded and executed by the user’s mobile device, is configured to cause the user’s mobile device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward” in claims 1 and 2 of the ’310 patent; limitations containing the

phrase "wherein the confirming step is performed before the submitting step and the user's mobile device receives an indication of a result of the processing the digital image of the check before the confirming step" in claim 9 of the '310 patent, limitations containing the phrase "the user's mobile device receiving an Indication of a result of the server's processing before confirming that the mobile check deposit can go forward" in claim 1 of the '095 patent, and limitations containing the phrase "the user's mobile device receives an Indication of a result of the processing the digital image of the check before the confirming step" in claim 15 of the '095 patent.

A person of ordinary skill in the art would have been motivated to combine such references because the references are all in the same field (image capture and processing) and address the same problem or objective (remotely depositing a check) in similar ways (using software on a user's device). *See e.g.*, Fellerman at [0018] ("In accordance with one embodiment of the present invention, a method for processing finance payments from a purchaser to a merchant in a transaction is contemplated. The method may begin with receiving images of a plurality of financial instruments. Each of the financial instruments may include a monetary value and at least one identifier associated with a financial account of the purchaser. The method may then continue with evaluating an approval status for the transaction based upon the account identifier on the financial instruments. Thereafter, the method may include transferring funds to the merchant. The funds may be transferred in response to the evaluation of the approval status. Additionally, the method may include the step of deducting funds from the financial account of the purchaser to the processor account at prescheduled intervals. An amount corresponding to the monetary value of a sequential one of the financial instruments may be deducted at each interval."), [0026] ("Generally, the remote terminal 26 is a data processing

apparatus for capturing, aggregating, and transmitting business transaction data to a remote system for completing payment. The remote terminal 26 includes a multifunction keypad 54, a screen 56 for displaying the results of the processed transactions and associated status messages, and a communications module 58. In addition to the foregoing, the remote terminal 26 may include a magnetic card/strip reader 36, as well as a printer 63 for reasons that will be described more fully below. By way of example only and not of limitation, the remote terminal 26 is an Omni Vx terminal from VeriFone, Inc. of San Jose, Calif., though any other suitable remote terminal device may be utilized. Depending upon the specific remote terminal 26, the above-described components may be internal thereto or connected to the terminal 26 as an external peripheral device."); Viera at Abstract ("Embodiments of the present invention include a computer-implemented method of performing a financial transaction on a network comprising storing a plurality of risk factors and associated severities in a web-enabled remote policy database, the severities defining actions taken when a specific risk factor is detected, storing a plurality of hold override policies that associate a deposit type with a hold type in the policy database, receiving deposit data including an image of a deposit check, applying the plurality of risk factors to the deposit data, and applying the plurality of hold override policies to the deposit data."); Cohen at [0022] ("Referring now to FIGS. 1 and 2, at the conclusion of either the scanning or data importation process step 202, in step 204 another process is triggered which performs OCR as well as parsing of the MICR Line information. In a subsequent step 206, the OCR and parsing processes return value data, including: (1) check amount; (2) bank routing number; (3) bank account number; and (4) check number. Significantly, these processes also return quantitative measurements, or values, representing the confidence level of the accuracy of the OCR results pertaining to each of the respective pieces of data. Subsequently, in step 208, a

remote operator at the local merchant site 100 reviews the images and data for each check in the batch. Significantly, the efficacy of this process is greatly enhanced by assigning color-coding in accordance with the respective confidence levels. Preferably, green, yellow and red colors are utilized to provide easy visual indication to the operator as to the confidence level attached to each data item. Subsequently, in step 210, and based upon a configuration setting in the profile that governs the particular batch, the operator is required to manually validate only those data items with a certain confidence level. Termed "Examination Rigor", this profile configuration value may be set to require that only low confidence data (i.e. marked red) need be inspected, or that both low and moderate confidence level data (i.e marked red or yellow) must be inspected, or that all data (i.e. even items marked green) must be inspected for errors. Upon completion of the review, the batch is ready for transmission to the DirectFed™ clearing system 118, or a like third party system. For convenience, all further description herein will refer to transmission to the DirectFed™ clearing system 118. In step 212, the image and data information is packaged at the client 100 for subsequent transmission, in step 214, to the DirectFed™ clearing system 118 via the Internet utilizing an encrypted web service."); Byrne at Abstract ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek -

Mobile Online Banking - Depositing your check explained at 0:09, 1:19; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone"), WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

As explained in Section V.B.1.m, it was well known to receive an indication of a result of processing on a customer's device to receive an indication of a result of processing. *See, e.g.*, Fellerman at [0043] ("Generally, the remote terminal 26 is a data processing apparatus for capturing, aggregating, and transmitting business transaction data to a remote system for completing payment. The remote terminal 26 includes a multifunction keypad 54, a screen 56 for displaying the results of the processed transactions and associated status messages, and a

communications module 58. In addition to the foregoing, the remote terminal 26 may include a magnetic card/strip reader 36, as well as a printer 63 for reasons that will be described more fully below. By way of example only and not of limitation, the remote terminal 26 is an Omni Vx terminal from VeriFone, Inc. of San Jose, Calif., though any other suitable remote terminal device may be utilized. Depending upon the specific remote terminal 26, the above-described components may be internal thereto or connected to the terminal 26 as an external peripheral device. “), [0048] (“All of the entered data is then verified in a check summary confirmation step 524 and a detail confirmation step 526. In one embodiment, the total of all the checks 24 presented as keyed into the remote terminal 26 in step 514 may confirmed. However, other related aggregate or summary data relating to the checks 24 may also be confirmed. Those having ordinary skill in the art will readily devise such alternative summary data. In the detail confirmation step 526, all or part of the entered data, such as the identification number, date of birth, issuing state, phone number, the check number for each of the scanned checks, and so forth may be verified.”), [0055] (“As indicated above, the remote terminal includes an integrated printer 63 to produce the receipt 110. The printer 63 may be any one of many widely utilized types, including thermal transfer, impact, and inkjet printers. As best illustrated in FIG. 8, the receipt 110 includes all of the pertinent information previously entered, as well additional details pertaining to the merchant 14. The receipt 110 may be segregated into multiple blocks, including a merchant identifier block 112 setting forth the name and address of the merchant 14. Additionally, a time and date stamp 114 is included that details the time and date the transaction was entered. A transaction details block 116 includes a unique identification number of the remote terminal 26, the batch number of the transaction, and the routing number 40 and account number 42 of the checks 24 presented. An itemized check listing 118 includes a column for an

item identification number, the amount and the check numbers 44 of the checks 24 presented, and the approval codes therefor. A total dollar amount for the entire transaction is also included. A notice block 120 highlights the most pertinent terms from the security agreement as another reminder of the same. Further information, such as advertising copy and the like, may be provided in a footer block 122. As indicated above, the receipt 110 is produced for both the purchaser 12 and the merchant 14, so each copy may be inscribed with a copy identifier 124. Although specifics of the receipt 110 have been described in detail, it will be appreciated by those having ordinary skill in the art that they are presented by way of example only and not of limitation. Any other suitable form and content of the receipt 110 may be substituted without departing from the scope of the present invention."); Viera at 7:33-64 ("FIGS. 5A-E are example graphical user interfaces illustrating a deposit transaction according to another embodiment of the present invention. FIG. 5A illustrates a menu of services 501 that is presented to the user. The user will select a service by touching a labeled button 502. If the user wants to end the transaction, he or she can touch the exit button 503. FIG. 5B illustrates a listing of the user's accounts 504. Each account is displayed with the account type 505, the balance 506, and the available funds 507. The user selects the account that will be receiving the deposit. FIG. 5C illustrates a check deposit 508. A check is inserted into an image scanner without an envelope. The account type 509 is displayed. Verification of check scan 510 is displayed. The user uses a keypad 511 to enter the deposit amount 512. An image 513 of the check is displayed to the user. FIG. 5D illustrates an appraisal of funds availability. The account type 515 is displayed. The deposit amount 516 is also displayed. The funds that are immediately available are displayed at 517A. The availability of the remaining funds is displayed at 517B. The user can reject the deposit by touching button 519 or button 518 to accept. If the deposit is accepted, the deposit

information is stored in a remote database that may be accessed by the user's financial institution. For example, after the transaction has completed, FIG. 5E illustrates instant online review of the deposit. A financial institution employee can review the details of the deposit and also see an image of the actual check. Previously, a check must be placed into an envelope for deposit. The envelope then sits in a depository until a teller can process the deposit and determine the availability. A user can then find out what the funds availability is through contacting the financial institution. The institution may also send a letter to the account holder's home with the funds availability."); Cohen at [0022] ("Referring now to FIGS. 1 and 2, at the conclusion of either the scanning or data importation process step 202, in step 204 another process is triggered which performs OCR as well as parsing of the MICR Line information. In a subsequent step 206, the OCR and parsing processes return value data, including: (1) check amount; (2) bank routing number; (3) bank account number; and (4) check number. Significantly, these processes also return quantitative measurements, or values, representing the confidence level of the accuracy of the OCR results pertaining to each of the respective pieces of data. Subsequently, in step 208, a remote operator at the local merchant site 100 reviews the images and data for each check in the batch. Significantly, the efficacy of this process is greatly enhanced by assigning color-coding in accordance with the respective confidence levels. Preferably, green, yellow and red colors are utilized to provide easy visual indication to the operator as to the confidence level attached to each data item. Subsequently, in step 210, and based upon a configuration setting in the profile that governs the particular batch, the operator is required to manually validate only those data items with a certain confidence level. Termed "Examination Rigor", this profile configuration value may be set to require that only low confidence data (i.e. marked red) need be inspected, or that both low and moderate confidence

level data (i.e. marked red or yellow) must be inspected, or that all data (i.e. even items marked green) must be inspected for errors. Upon completion of the review, the batch is ready for transmission to the DirectFed™ clearing system 118, or a like third party system. For convenience, all further description herein will refer to transmission to the DirectFed™ clearing system 118. In step 212, the image and data information is packaged at the client 100 for subsequent transmission, in step 214, to the DirectFed™ clearing system 118 via the Internet utilizing an encrypted web service."), [0053] ("Step 14: The DirectFed™ reporting system generates a report (in Skyline Data proprietary format) for delivery to the merchant. This report gives the total deposit of the merchant, subtotals by bank, subtotals by account number, subtotals by merchant location, as well as detailed information from each check deposited."); Byrne at [0189]-[0190] ("An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent. The acknowledgment is uploaded to the user's client, the content of which depends on the client options."); ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09, 1:19; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone

will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone"); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

A person of ordinary skill in the art would have been motivated to incorporate these teachings into Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU because doing so would have kept the user appraised of the status of the check deposit, including whether or not the image processing was successful. Indicating success to the user after processing, as opposed to waiting until the deposit had been submitted/approved, would have provided the user with relevant check processing information without having to wait for the longer check deposit/approval process to occur. Moreover, to the extent that USAA interprets this limitation to only require receiving an indication of a result of processing in software and not displaying such indication to a user, providing such indication would allow the user (or device) to confirm going forward with deposit for the reasons discussed in Section V.B.3.b.vi knowing that the image was of sufficient quality for deposit. References such as Cohen, Fellerman, Viera, Byrne, ImageNet, and/or WVUFCU disclose the benefits of incorporating receiving an

indication of a result of processing into such systems. *See, e.g.*, Fellerman at [0043] ("Generally, the remote terminal 26 is a data processing apparatus for capturing, aggregating, and transmitting business transaction data to a remote system for completing payment. The remote terminal 26 includes a multifunction keypad 54, a screen 56 for displaying the results of the processed transactions and associated status messages, and a communications module 58. In addition to the foregoing, the remote terminal 26 may include a magnetic card/strip reader 36, as well as a printer 63 for reasons that will be described more fully below. By way of example only and not of limitation, the remote terminal 26 is an Omni Vx terminal from VeriFone, Inc. of San Jose, Calif., though any other suitable remote terminal device may be utilized. Depending upon the specific remote terminal 26, the above-described components may be internal thereto or connected to the terminal 26 as an external peripheral device. "), [0048] ("All of the entered data is then verified in a check summary confirmation step 524 and a detail confirmation step 526. In one embodiment, the total of all the checks 24 presented as keyed into the remote terminal 26 in step 514 may confirmed. However, other related aggregate or summary data relating to the checks 24 may also be confirmed. Those having ordinary skill in the art will readily devise such alternative summary data. In the detail confirmation step 526, all or part of the entered data, such as the identification number, date of birth, issuing state, phone number, the check number for each of the scanned checks, and so forth may be verified."), [0055] ("As indicated above, the remote terminal includes an integrated printer 63 to produce the receipt 110. The printer 63 may be any one of many widely utilized types, including thermal transfer, impact, and inkjet printers. As best illustrated in FIG. 8, the receipt 110 includes all of the pertinent information previously entered, as well additional details pertaining to the merchant 14. The receipt 110 may be segregated into multiple blocks, including a merchant identifier block 112 setting forth the name

and address of the merchant 14. Additionally, a time and date stamp 114 is included that details the time and date the transaction was entered. A transaction details block 116 includes a unique identification number of the remote terminal 26, the batch number of the transaction, and the routing number 40 and account number 42 of the checks 24 presented. An itemized check listing 118 includes a column for an item identification number, the amount and the check numbers 44 of the checks 24 presented, and the approval codes therefor. A total dollar amount for the entire transaction is also included. A notice block 120 highlights the most pertinent terms from the security agreement as another reminder of the same. Further information, such as advertising copy and the like, may be provided in a footer block 122. As indicated above, the receipt 110 is produced for both the purchaser 12 and the merchant 14, so each copy may be inscribed with a copy identifier 124. Although specifics of the receipt 110 have been described in detail, it will be appreciated by those having ordinary skill in the art that they are presented by way of example only and not of limitation. Any other suitable form and content of the receipt 110 may be substituted without departing from the scope of the present invention."); Viera at 7:33-64 ("FIGS. 5A-E are example graphical user interfaces illustrating a deposit transaction according to another embodiment of the present invention. FIG. 5A illustrates a menu of services 501 that is presented to the user. The user will select a service by touching a labeled button 502. If the user wants to end the transaction, he or she can touch the exit button 503. FIG. 5B illustrates a listing of the user's accounts 504. Each account is displayed with the account type 505, the balance 506, and the available funds 507. The user selects the account that will be receiving the deposit. FIG. 5C illustrates a check deposit 508. A check is inserted into an image scanner without an envelope. The account type 509 is displayed. Verification of check scan 510 is displayed. The user uses a keypad 511 to enter the deposit amount 512. An image 513 of the

check is displayed to the user. FIG. 5D illustrates an appraisal of funds availability. The account type 515 is displayed. The deposit amount 516 is also displayed. The funds that are immediately available are displayed at 517A. The availability of the remaining funds is displayed at 517B. The user can reject the deposit by touching button 519 or button 518 to accept. If the deposit is accepted, the deposit information is stored in a remote database that may be accessed by the user's financial institution. For example, after the transaction has completed, FIG. 5E illustrates instant online review of the deposit. A financial institution employee can review the details of the deposit and also see an image of the actual check. Previously, a check must be placed into an envelope for deposit. The envelope then sits in a depository until a teller can process the deposit and determine the availability. A user can then find out what the funds availability is through contacting the financial institution. The institution may also send a letter to the account holder's home with the funds availability."); Cohen at [0042] ("Step 3: The scanner processes each check, including the printed endorsement on the back of each check, reading the magnetic MICR line, and providing an image of the front and back of the each check. This information is stored on disk and the process continues until the operator determines there are no more checks to scan, or the maximum number of checks in the batch, as defined in the profile, has been reached. At that time the operator can review each check and, optionally, can remove and rescan checks having potentially inadequate or poor quality images, or rotate checks scanned, for example, in an upside-down orientation. When the operator is satisfied, the batch is concluded, whereupon the images are stored in the local client image repository and the information about the batch and each check is recorded in the local client database. When the scanner captures an image of an item, a security hash value is simultaneously created and stored along with it. This hash value is used to ensure the integrity of the image and prevent any attempts to alter it from the instant it is

captured by the scanner and stored on the hard drive of the client PC, until it is packaged and uploaded to the Skyline Data DirectFed™ clearing system."), [0022] ("Referring now to FIGS. 1 and 2, at the conclusion of either the scanning or data importation process step 202, in step 204 another process is triggered which performs OCR as well as parsing of the MICR Line information. In a subsequent step 206, the OCR and parsing processes return value data, including: (1) check amount; (2) bank routing number; (3) bank account number; and (4) check number. Significantly, these processes also return quantitative measurements, or values, representing the confidence level of the accuracy of the OCR results pertaining to each of the respective pieces of data. Subsequently, in step 208, a remote operator at the local merchant site 100 reviews the images and data for each check in the batch. Significantly, the efficacy of this process is greatly enhanced by assigning color-coding in accordance with the respective confidence levels. Preferably, green, yellow and red colors are utilized to provide easy visual indication to the operator as to the confidence level attached to each data item. Subsequently, in step 210, and based upon a configuration setting in the profile that governs the particular batch, the operator is required to manually validate only those data items with a certain confidence level. Termed "Examination Rigor", this profile configuration value may be set to require that only low confidence data (i.e. marked red) need be inspected, or that both low and moderate confidence level data (i.e marked red or yellow) must be inspected, or that all data (i.e. even items marked green) must be inspected for errors. Upon completion of the review, the batch is ready for transmission to the DirectFed™ clearing system 118, or a like third party system. For convenience, all further description herein will refer to transmission to the DirectFed™ clearing system 118. In step 212, the image and data information is packaged at the client 100 for subsequent transmission, in step 214, to the DirectFed™ clearing system 118 via the Internet

utilizing an encrypted web service."); Byrne at [0018]-[0032] ("It further would be advantageous to provide a system and method that attains the following business goals: … provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and provide a same and/or better than response time."); ImageNet, J&B Software Selects Mitek at 1 ("J&B Software Selects ImageNetMobile Deposit to Extend Remote Imaging Capture Functionality to Mobile Phones"), J&B to Showcase Mobile Deposit at 1 ("J&B Software to Showcase Mobile Deposit, eCapture and Electronic Clearing Suite at BAI TransPayConference"); Mobile Marketer at Page 1; Mitek - Mobile Online Banking - Depositing your check explained at 0:09, 1:19; NCR Unleashes Remote Check Deposit by Using Mobile Phone at Page 1 ("NCR unleashes remote check deposit by using mobile phone camera"); Mitek – RDM-to-Target-Small-Businesses-with-ImageNet-Mobil at Page 1 ("RDM to Target Small Businesses with ImageNet Mobile Deposit ™ Application for Blackberry and iPhone"); Remote Deposit Capture Goes Mobile Phone Camera at Page 1 ("Remote Deposit Capture Goes Mobile: Mitek Systems Launches First Mobile Check Deposit and Bill Pay Application"); Your iPhone will be your wallet - 9to5Mac at Page 1 ("We've mused on this before, but things have taken another step forward with news of Mitek's Mobile Deposit (read, banking) application for the iPhone"); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the

whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

It would have therefore been obvious to a person of ordinary skill in the art to receive an indication of a result of processing, for example as taught in Cohen, Fellerman, Viera, Byrne, ImageNet, and/or WVUFCU. Doing so would have involved simply combining prior art elements/techniques (e.g., check deposit processing steps) according to known devices/methods to yield predicable results (e.g., processing a check for deposit using its image).

A person of ordinary skill in the art would have had a reasonable expectation of success. For example, implementing a mobile device that receives an indication of a result of processing would have simply involved writing computer code to provide and then receive such indication after processing occurs. The references use similar well-known components such that combining the references to receive an indication of a result of processing in the proposed combination(s) would have been a routine, predictable design for a person of ordinary skill in the art and would not have involved any undue experimentation.

x. <u>It would have been obvious to a POSITA to update an account balance ('310 and '095 Patents)</u>

To the extent USAA contends that the prior art identified in Section V.A does not disclose updating an account balance using a bank computer, a POSITA would have found such feature obvious in view of Cohen, Fellerman, Viera, Byrne, Rodriguez, CAM, and/or Garcia, at least under USAA's interpretation of the term. As an illustrative, non-limiting example, a person of ordinary skill in the art would have been motivated to combine the teachings of references such as Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (which disclose, at least in one embodiment, depositing a check using an image captured by a mobile device) with the general knowledge and skill of a person skilled in the art and optionally

references such as Cohen, Fellerman, Viera, Byrne, Rodriguez, CAM, and/or Garcia (which disclose a computer updating a balance as a part of check deposit) to complete the check deposit. Such a combination would render obvious, for example, limitations containing the phrase "updating, by a [bank] computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device" in claims 1, 2, 5, 6, 8, 9, 10, 11, 14, 15, of the '310 patent; limitations containing the phrase "a computer comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the computer to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device" in claims 1, 18 of the '095 patent; and limitations containing the phrase "a [bank] computer programmed to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device" in claims 9, 26 of the '095 patent.

A person of ordinary skill in the art would have been motivated to combine such references because the references are all in the same field (image capture and processing) and address the same problem or objective (remotely depositing a check) in similar ways (using an image of the check and computers). *See e.g.*, Rodriguez at [0010] ("The problems identified above are in large part addressed by methods, systems, and media to process and clear a check. One embodiment provides a method for generating an e-check. The method generally includes scanning a check to create an image of the check in response to receiving a check as payment for a transaction; entering an amount of currency represented by the check into a point of sale terminal; identifying check information that describes a bank and a bank account, wherein the

check comprises the check information; and generating a check object based upon the image, the amount, and the check information."); Fellerman at [0018] ("In accordance with one embodiment of the present invention, a method for processing finance payments from a purchaser to a merchant in a transaction is contemplated. The method may begin with receiving images of a plurality of financial instruments. Each of the financial instruments may include a monetary value and at least one identifier associated with a financial account of the purchaser. The method may then continue with evaluating an approval status for the transaction based upon the account identifier on the financial instruments. Thereafter, the method may include transferring funds to the merchant. The funds may be transferred in response to the evaluation of the approval status. Additionally, the method may include the step of deducting funds from the financial account of the purchaser to the processor account at prescheduled intervals. An amount corresponding to the monetary value of a sequential one of the financial instruments may be deducted at each interval."); Viera at Abstract ("Embodiments of the present invention include a computer-implemented method of performing a financial transaction on a network comprising storing a plurality of risk factors and associated severities in a web-enabled remote policy database, the severities defining actions taken when a specific risk factor is detected, storing a plurality of hold override policies that associate a deposit type with a hold type in the policy database, receiving deposit data including an image of a deposit check, applying the plurality of risk factors to the deposit data, and applying the plurality of hold override policies to the deposit data."); Cohen at [0025] ("The DirectFed™ system takes advantage of the new rules provided by the Check 21 act. It allows merchants to deposit checks without having to deliver the physical check to a bank. The DirectFed™ clearing system has been approved by the Federal Reserve Bank for the delivery of checks via the Check 21 system. The checks, preferably

delivered/routed via a third party system, such as the DirectFed™ clearing system 118, are directly deposited with the Federal Reserve Bank 130 (or Endpoint Exchange 132), completely bypassing the BOFD 128. There are a host of benefits and advantages provided by the system and method of the present invention, including, for example, the following: (1) merchant employees do not have to leave the office to deposit checks; (2) manual deposit tickets do not have to be created; (3) photocopies of checks do not have to be created, since the images of the checks are retained and available on the client system; (4) for large volume deposits, armored guards/vehicles are not required, since the physical checks are never transported to the bank; (5) rather than having paper checks delivered to the bank and then manually processed at the bank, the check images are electronically delivered to the Federal Reserve Bank, or Endpoint Exchange, for credit to the BOFD, resulting in greatly expedited availability of funds; (6) due to the increased clearance speed, returns are processed faster, giving merchants a greater opportunity to recover funds; (7) upon request of the merchant, checks can be automatically separated into those eligible for ACH and those which are not eligible, so that ACH eligible checks can be processed by ACH and the others via Check 21, preventing the requirement of the merchant manually separating checks before processing via ACH. With the addition of coupon processing (further described herein below), or without coupon processing (depending upon the required data), automated import into client Accounts Receivable (A/R) and other systems can be easily accommodated. This can result in an enormous saving of time, and elimination of many common keyboard errors. The BOFD does not need to have a physical location near the remote merchant site. This allows a bank to expand the geographic coverage of its depositor. Similarly, a merchant with multiple locations can consolidate its accounts at a single bank, even for those locations that are not located near a bank branch. Banks that clear checks through other banks, or

credit unions clearing through corporate credit unions, can bypass the intermediary and clear directly through the Federal Reserve Bank. This saves time and the cost of physically delivering the checks."); Byrne at Abstract ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); CAM (Parikh 2005a) at 1 ("In this paper we introduce mobile document processing as an accessible and locally harmonious way to provide information services to the developing world [2]. In our system a camera-equipped mobile phone is used for image capture and data entry. Document interaction is specified using CamShell - a narrative document-embedded programming language that can allow developers to add interactive audio, data entry, validation, processing, and networking instructions to otherwise inert paper documents.").

As explained in Section V.B.1.n, it was well known to use a bank computer to update a balance in a remote deposit system to complete the deposit. *See, e.g.*, Rodriguez at [0034] ("Once e-check clearer 142 determines that the funds are available in purchaser's account 144, e-check clearer 142 may transmit a confirmation to POS terminal 130 and transfer the funds to merchant's account 152 via LAN/WAN 160. In other embodiments, purchaser's bank 140 may transfer the funds to merchant's account 152 and merchant's bank 150 may confirm the receipt of the funds with POS terminal 130."), [0006] ("The merchant's bank is disadvantaged because the merchant's bank will show the balance of the check in the merchant's bank account for the 'float' period. Further, the merchant is disadvantaged because the merchant will not know during the delay, possibly for several days, whether the check is actually good, i.e., whether the

purchaser actually has sufficient funds available for the transaction, and, even if the purchaser does have the funds available, the funds that show in the merchant's account may be placed on hold by the merchant's bank until they are actually received or for some standard number of days to make sure that the funds are received before being available to the merchant."); Fellerman at [0061] ("In further detail, the images of the checks 24 received from the remote terminal 26 or the check scanner 28 as described above are stored at the central payment processor 52. At the specified 30-day intervals, the images of the checks 24 are transmitted as remote deposits under the Check 21 framework into the processor account 108, at which time the bank initiates a request with the purchaser's bank 20 for the transfer of the appropriate amount of funds therefrom. It is also contemplated that third party processors other than the banks may handle the transfers. As previously noted, it is possible for a copy of the check 24 to function as an ACH debit authorization by way of the security agreement. In this regard, the central payment processor 52 is understood to originate an ACH credit thereto that is to be deducted from the purchaser's bank account 20. The ACH credit is likewise originated at 30-day intervals according to the predefined schedule described previously. To the extent that any check 24 fails to clear, the central payment processor 52 is responsible therefor, as the reimbursement obligation to the merchant 14 remains. However, because the central payment processor 52 retains interest in the purchaser's repayment obligations, it may independently initiate a collection action against the purchaser 10 without involvement of the merchant 14."), [0016] ("One alternative to short-term financing with credit cards is a payment structure in which future dated checks are sequentially processed on the dates specified thereby, with representative examples being disclosed in U.S. Pat. App. Pub. No. 2006/0015428 and U.S. Pat. App. Pub. No. 2006/0015427, both to Friedman. In further detail, Friedman contemplates a check writer

presenting one or more future-dated checks to a merchant, followed by a verification of the presenter's check writing history. Furthermore, the balance history of the presenter's account may also be evaluated to determine an appropriate monthly payment amount or total financing amount. If these verifications are acceptable, the merchant assigns all of the checks to a guarantee entity, and immediately receives the total amount for all of the checks presented, minus a service charge. The check guaranty entity then deposits the checks on the dates specified in each of the future-dated checks."); Viera at 7:1-32 ("FIG. 4 illustrates a method of performing a transaction on a network according to one embodiment of the present invention. FIG. 4 illustrates a method of performing a deposit transaction. At step 401, a user inserts a check into an image scanner at a kiosk. At step 402, the back and front of the check are scanned and the images are stored (e.g. bitmaps). The magnetic encoding ("MICR") at the bottom of the check is also scanned. The user enters the amount of the deposit using a touchscreen keypad at step 403. At step 404, the bitmap images and MICR obtained from step 402 are passed to an optical character recognition ("OCR") program. OCR will perform several functions. It will re-read the MICR using optical instead of magnetic techniques. It will detect the check amount, payee name, and check number. It will detect the imprinted name of the paying financial institution. It will determine if a signature is present on the front of the check. It will also determine if a signature is present in the endorsement section on the back of the check. A deposit type is determined at step 405 using the information obtained from step 404. In making this determination, risk policies and routing group information from database tables 406 and 407 are used. If deposit is rejected at step 408, the check is returned to the user. If the deposit is accepted, it will be examined for hold override policies using policy information stored in database 411. Funds availability is calculated and presented to the user at step 412. If the user rejects the deposit at step 413, then the check

will be returned at step 414. If the user elects to proceed, deposit information will be printed on the check and stored in a secure internal bin at step 415. At step 416, the check images and deposit information will be posted through the shared branching network 417 to be received by the appropriate financial institution."), 7:32-64 ("FIGS. 5A-E are example graphical user interfaces illustrating a deposit transaction according to another embodiment of the present invention. FIG. 5A illustrates a menu of services 501 that is presented to the user. The user will select a service by touching a labeled button 502. If the user wants to end the transaction, he or she can touch the exit button 503. FIG. 5B illustrates a listing of the user's accounts 504. Each account is displayed with the account type 505, the balance 506, and the available funds 507. The user selects the account that will be receiving the deposit. FIG. 5C illustrates a check deposit 508. A check is inserted into an image scanner without an envelope. The account type 509 is displayed. Verification of check scan 510 is displayed. The user uses a keypad 511 to enter the deposit amount 512. An image 513 of the check is displayed to the user. FIG. 5D illustrates an appraisal of funds availability. The account type 515 is displayed. The deposit amount 516 is also displayed. The funds that are immediately available are displayed at 517A. The availability of the remaining funds is displayed at 517B. The user can reject the deposit by touching button 519 or button 518 to accept. If the deposit is accepted, the deposit information is stored in a remote database that may be accessed by the user's financial institution. For example, after the transaction has completed, FIG. 5E illustrates instant online review of the deposit. A financial institution employee can review the details of the deposit and also see an image of the actual check. Previously, a check must be placed into an envelope for deposit. The envelope then sits in a depository until a teller can process the deposit and determine the availability. A user can then find out what the funds availability is through contacting the financial institution. The institution

may also send a letter to the account holder's home with the funds availability."); Cohen at [0051] ("Step 12: Upon acceptance of the X937 file by the Federal Reserve Bank, further processing of the checks is performed by the Federal Reserve Bank. The Federal Reserve Bank performs a net settlement of funds to the BOFDs."), [0025] ("The DirectFed™ system takes advantage of the new rules provided by the Check 21 act. It allows merchants to deposit checks without having to deliver the physical check to a bank. The DirectFed™ clearing system has been approved by the Federal Reserve Bank for the delivery of checks via the Check 21 system. The checks, preferably delivered/routed via a third party system, such as the DirectFed™ clearing system 118, are directly deposited with the Federal Reserve Bank 130 (or Endpoint Exchange 132), completely bypassing the BOFD 128. There are a host of benefits and advantages provided by the system and method of the present invention, including, for example, the following: (1) merchant employees do not have to leave the office to deposit checks; (2) manual deposit tickets do not have to be created; (3) photocopies of checks do not have to be created, since the images of the checks are retained and available on the client system; (4) for large volume deposits, armored guards/vehicles are not required, since the physical checks are never transported to the bank; (5) rather than having paper checks delivered to the bank and then manually processed at the bank, the check images are electronically delivered to the Federal Reserve Bank, or Endpoint Exchange, for credit to the BOFD, resulting in greatly expedited availability of funds; (6) due to the increased clearance speed, returns are processed faster, giving merchants a greater opportunity to recover funds; (7) upon request of the merchant, checks can be automatically separated into those eligible for ACH and those which are not eligible, so that ACH eligible checks can be processed by ACH and the others via Check 21, preventing the requirement of the merchant manually separating checks before processing via ACH. With the

addition of coupon processing (further described herein below), or without coupon processing (depending upon the required data), automated import into client Accounts Receivable (A/R) and other systems can be easily accommodated. This can result in an enormous saving of time, and elimination of many common keyboard errors. The BOFD does not need to have a physical location near the remote merchant site. This allows a bank to expand the geographic coverage of its depositor. Similarly, a merchant with multiple locations can consolidate its accounts at a single bank, even for those locations that are not located near a bank branch. Banks that clear checks through other banks, or credit unions clearing through corporate credit unions, can bypass the intermediary and clear directly through the Federal Reserve Bank. This saves time and the cost of physically delivering the checks."); Garcia at 4: 11-18 ("The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations."); 7:22-8:8 ("The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution."); 11: 15-19 ("The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems)."); 12:17-22 ("The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account."); CAM (Parikh 2007) at 75 ("The resulting

reports and financial statements are printed from a secure access point, or faxed from the Federation head office to the field office, where they are picked up by the group or delivered by the staff. Reports can contain interactive CAM content, for example, for accessing up-to-date account balances. These reports are filed in a set of folders, taking the place of the shared notebooks. They are used by the group for monitoring their portfolio and applying for loans and other services. Paper remains the main display and storage medium for SHGs and their members.").

A person of ordinary skill in the art would have been motivated to incorporate these teachings into Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU because doing so would have been the natural next step in completing check deposit. Indeed, updating a balance using a computer to reflect deposited funds from a check long predated remote deposit systems. Thus, a person of skill in the art would have recognized that when performing check deposit using a mobile device, it would have been necessary to update an account balance to reflect the deposited funds just as it had been done for checks deposited with checks deposited with a bank teller or using an ATM. Indeed, it would have been inefficient to update the account balance using anything other than a bank computer.

It would have therefore been obvious to a person of ordinary skill in the art to use a bank computer to update a balance, for example as taught in Cohen, Fellerman, Viera, Byrne, Rodriguez, CAM, and/or Garcia. Doing so would have involved simply combining prior art elements/techniques (e.g., updating a balance) according to known devices/methods (e.g., with a computer) to yield predicable results (e.g., completing the deposit).

A person of ordinary skill in the art would have had a reasonable expectation of success. The entire goal of check deposit—remote or otherwise—is to update the payor and payees

account balances to reflect a transfer of funds. Banks had been using computers to keep track of account balances for many years, and using such computers to update account balances based on check deposit was the industry norm. Moreover, using a bank computer to update a balance would have simply involved writing computer code to change a balance responsive to successful check deposit. The references use similar well-known components such that combining the references to use a bank computer to update a balance in the proposed combination(s) would have been a routine, predictable design for a person of ordinary skill in the art and would not have involved any undue experimentation.

xi. It would have been obvious to a POSITA to convert the format of an image of a check (’310 and ’095 Patents)

To the extent USAA contends that the prior art identified in Section V.A does not disclose converting the format of an image in a check, a POSITA would have found such feature obvious in view of Cohen, Byrne, Fellerman, Nepomniachtchi, and/or WVUFCU, at least under USAA’s interpretation of the term. As an illustrative, non-limiting example, a person of ordinary skill in the art would have been motivated to combine the teachings of references such as Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU (which disclose, at least in one embodiment, depositing a check using an image captured by a mobile device) with references such as Cohen, Byrne, Fellerman, Nepomniachtchi, and/or WVUFCU (which disclose converting an image of a check to a different format) to produce a check image that complies with the relevant U.S. deposit standards. Such a combination would render obvious, for example, limitations containing the phrase “wherein the digital image is a modified version of the captured video frame” in claims 7, 24 of the ’095 patent.

A person of ordinary skill in the art would have been motivated to combine such references because the references are all in the same field (image capture and processing) and

address the same problem or objective (check deposit) in similar ways (using the image of a check). *See e.g.*, Fellerman at [0018] ("In accordance with one embodiment of the present invention, a method for processing finance payments from a purchaser to a merchant in a transaction is contemplated. The method may begin with receiving images of a plurality of financial instruments. Each of the financial instruments may include a monetary value and at least one identifier associated with a financial account of the purchaser. The method may then continue with evaluating an approval status for the transaction based upon the account identifier on the financial instruments. Thereafter, the method may include transferring funds to the merchant. The funds may be transferred in response to the evaluation of the approval status. Additionally, the method may include the step of deducting funds from the financial account of the purchaser to the processor account at prescheduled intervals. An amount corresponding to the monetary value of a sequential one of the financial instruments may be deducted at each interval."); Cohen at [0025] ("The DirectFed™ system takes advantage of the new rules provided by the Check 21 act. It allows merchants to deposit checks without having to deliver the physical check to a bank. The DirectFed™ clearing system has been approved by the Federal Reserve Bank for the delivery of checks via the Check 21 system. The checks, preferably delivered/routed via a third party system, such as the DirectFed™ clearing system 118, are directly deposited with the Federal Reserve Bank 130 (or Endpoint Exchange 132), completely bypassing the BOFD 128. There are a host of benefits and advantages provided by the system and method of the present invention, including, for example, the following: (1) merchant employees do not have to leave the office to deposit checks; (2) manual deposit tickets do not have to be created; (3) photocopies of checks do not have to be created, since the images of the checks are retained and available on the client system; (4) for large volume deposits, armored

guards/vehicles are not required, since the physical checks are never transported to the bank; (5) rather than having paper checks delivered to the bank and then manually processed at the bank, the check images are electronically delivered to the Federal Reserve Bank, or Endpoint Exchange, for credit to the BOFD, resulting in greatly expedited availability of funds; (6) due to the increased clearance speed, returns are processed faster, giving merchants a greater opportunity to recover funds; (7) upon request of the merchant, checks can be automatically separated into those eligible for ACH and those which are not eligible, so that ACH eligible checks can be processed by ACH and the others via Check 21, preventing the requirement of the merchant manually separating checks before processing via ACH. With the addition of coupon processing (further described herein below), or without coupon processing (depending upon the required data), automated import into client Accounts Receivable (A/R) and other systems can be easily accommodated. This can result in an enormous saving of time, and elimination of many common keyboard errors. The BOFD does not need to have a physical location near the remote merchant site. This allows a bank to expand the geographic coverage of its depositor. Similarly, a merchant with multiple locations can consolidate its accounts at a single bank, even for those locations that are not located near a bank branch. Banks that clear checks through other banks, or credit unions clearing through corporate credit unions, can bypass the intermediary and clear directly through the Federal Reserve Bank. This saves time and the cost of physically delivering the checks."); Byrne at Abstract ("A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided."); WVUFCU (iPhoneApp) at 1

("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

As explained in Section V.B.1.o, it was well known to reformat a check image in a check deposit system to ensure compliance with the relevant U.S. check deposit standards and practices under Check 21, including, for example, the Federal Reserve's DSTU X9.37- 2003 format, which was known to require a black and white (bitonal) TIFF format. *See, e.g.*, Fellerman at [0037] ("The merchant 14 may operate a remote terminal 26 that includes a check scanner 28 that is capable of capturing the images of the checks 24. In capturing the images, for example, a Tagged Image File Format (TIFF) file may be created. Any one of widely available check scanners 28 that can be controlled from the remote terminal 26 may be utilized, such as the EC5000 series and EC6000 series imagers from RDM Corporation of Waterloo, Ontario, Canada."), [0038] ("FIGS. 4 a and 4 b are representative examples of the front and back, respectively, of the check 24 and it is understood that the check scanner 28 produces data representative thereof for electronic transmission. Preferably, the format of the data produced by the check scanner 28 conforms to the X9.90 standard that sets forth the construction, layout, data elements, data content, and printing specifications for the truncated check 24."); Cohen at [0026] ("In contrast to systems and methods of the prior art, the system and methods of the present

invention enable and facilitate deliver directly to the Federal Reserve Bank. Existing systems require delivery to the BOFD, which then processes the checks for clearance. Furthermore, the system and methods of the present invention enable automatic determination as to whether to process a check by ACH or Check 21. The DirectFed™ clearing system is capable of creating multiple Check 21 compliant X937 files, each file containing multiple cash letters and bundles. This allows delivery of checks for multiple banks to multiple Federal Reserve Banks for multiple accounts."), claim 1 ("a second subsystem at said client location, in secure communication with said first subsystem, configured for receiving checking document data, converting the checking document data into a format in conformance with Check 21 processing requirements, and securely transmitting said formatted data to said first subsystem for direct routing to said Federal Reserve bank."); Byrne at [0088] ("This file format is based on the ANSI X9.37 DSTU—2003 "Specifications for Electronic Exchange of Check and Image Data" draft standard with some modifications to accommodate the processing of credits, such as customer Direct Deposit Account (DDA) deposits and adjustments, and to augment sender source identification and file balancing controls."); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

A person of ordinary skill in the art would have been motivated to incorporate these teachings into Garcia, Acharya, Nepomniachtchi, Lin, Rodriguez, ImageNet, and WVUFCU because doing so would have provided a usable check image for processing by the Federal Reserve and other entities involved in deposit. When choosing where and when to convert the format of the image, a person of ordinary skill in the art would have found it obvious to do so on the user's device, which would reduce the size of the image being transmitted, thereby saving bandwidth. References such as Cohen, Byrne, Fellerman, Nepomniachtchi, and/or WVUFCU disclose the benefits of changing the format of a check image into such systems. *See, e.g.*, Fellerman at [0038] ("FIGS. 4 a and 4 b are representative examples of the front and back, respectively, of the check 24 and it is understood that the check scanner 28 produces data representative thereof for electronic transmission. Preferably, the format of the data produced by the check scanner 28 conforms to the X9.90 standard that sets forth the construction, layout, data elements, data content, and printing specifications for the truncated check 24."); [0021] ("As an improvement over conventional check processing and an alternative to ACH, the United States Congress enacted the Check Clearing for the 21st Century Act, also known as Check 21, for facilitating the electronic processing of checks. Check 21 allows banks to capture an image of the front and back of the check in a process referred to as truncation. The captured check image data, which serves as a legal substitute for a physical copy of the check, is then exchanged between member banks, savings and loans, credit unions, servicers, clearinghouses, and the Federal Reserve banks for payment processing. In addition, due to the widespread availability of scanners and check reading devices, it is possible for merchants to deposit checks remotely without having to be physically delivered to the bank. Because the need for transporting checks is eliminated, efficiency gains and cost reductions can be achieved, and the financial system can

remain operational even in emergencies that affect transportation infrastructure."); Cohen at [0026] ("In contrast to systems and methods of the prior art, the system and methods of the present invention enable and facilitate deliver directly to the Federal Reserve Bank. Existing systems require delivery to the BOFD, which then processes the checks for clearance. Furthermore, the system and methods of the present invention enable automatic determination as to whether to process a check by ACH or Check 21. The DirectFed™ clearing system is capable of creating multiple Check 21 compliant X937 files, each file containing multiple cash letters and bundles. This allows delivery of checks for multiple banks to multiple Federal Reserve Banks for multiple accounts."), [0028] ("Checks processed via Check 21 are sent to the Federal Reserve Bank via secure transport. One or more X937 files are created as required. Each X937 file is created and delivered via secure transport to a specific branch of the Federal Reserve Bank. The rules engine is used to determine which branch of the Federal Reserve Bank to deliver a check to. The routing number of the BOFD is used to make this determination. Each X937 file for a specific Federal Reserve Bank branch is composed of one or more cash letters. The rules engine determines the number of cash letters in an X937 file based upon analysis of the Banks of First Deposit. Each X937 file is created utilizing proprietary Skyline Data system software created to meet the specifications set by the Federal Reserve Bank. Each X937 files is delivered to the Federal Reserve Bank via a secure transport method approved by the Federal Reserve Bank."); Byrne at [0088] ("This file format is based on the ANSI X9.37 DSTU—2003 "Specifications for Electronic Exchange of Check and Image Data" draft standard with some modifications to accommodate the processing of credits, such as customer Direct Deposit Account (DDA) deposits and adjustments, and to augment sender source identification and file balancing controls."); WVUFCU (iPhoneApp) at 1 ("iPhone App. iDeposit allows WV United

FCU members to deposit paper checks electronically using your iPHone"); Finovate Blog at 2 ("But somehow they were able to pull off something that no other financial institution has yet to accomplish, accepting paper check deposits via a native iPhone application . . . .The application was added to the iPhone AppStore on July 4, and the CU wasted no time in heralding the innovation with a homepage banner."); An Eye for Technology at 1 ("With the iPhone RDC strategy, that was something we had thought about years before the whole Apple app phenomenon happened. But at the time there was no secure way to send images on a phone, so Apple helped us out with that.").

It would have therefore been obvious to a person of ordinary skill in the art to convert the format of the check image, for example as taught in Cohen, Byrne, Fellerman, Nepomniachtchi, and/or WVUFCU. Doing so would have involved simply combining prior art elements/techniques (e.g., image format conversions) according to known devices/methods to yield predicable results (e.g., a check image that complies with relevant standards) and the simple substitution of one known element (e.g., one image format) for another (e.g., an X9-compliant format) to obtain predictable results (e.g., a check image that complies with relevant standards).

A person of ordinary skill in the art would have had a reasonable expectation of success. For example, both image conversion and the X9 standard had been known for many years. Therefore, converting an image to comply with the X9 standard would have been well within the level of ordinary skill in the art and would have simply involved writing computer code to perform the conversion. The references use similar well-known components such that combining the references to convert the format of the check image in the proposed

combination(s) would have been a routine, predictable design for a person of ordinary skill in the art and would not have involved any undue experimentation.

#### 4. No Objective Indicia of Non-Obviousness

USAA bears the burden of coming forward with evidence regarding objective indicia of non-obvious and their purported nexus to the asserted claims. *ZUP, LLC v. Nash Mfg., Inc.*, 896 F.3d 1365, 1373 (Fed. Cir. 2018). To date, USAA has not identified any evidence nor made any arguments pertaining to relevant objective indicia of non-obviousness.

Regardless of any purported objective indicia of non-obviousness, any such objective indicia cannot overcome the prima facie showing of obviousness, as demonstrated by the teachings of the prior art and combinations of prior art described above. Additionally, the contemporaneous invention of the purported subject matter of the Asserted Claims by others in the field, without reliance on the Asserted Patents, in fact serve as objective evidence of obviousness.

USAA has failed to show that any objective indicia of non-obviousness overcome the compelling prima facie showing that the Asserted Claims of the Asserted Patents would have been obvious. Regions reserves the right to amend and/or supplement these Invalidity Contentions to the extent USAA later presents new argument.

## VI. CONTENTIONS UNDER 35 U.S.C. § 112

To comply with the written description requirement of 35 U.S.C. § 112, a written description must "reasonably convey[] to those skilled in the art that the inventor had possession of the claimed subject matter as of the filing date." *Ariad Pharm., Inc. v. Eli Lilly & Co.*, 598 F.3d 1336, 1351 (Fed. Cir. 2010) (en banc) (citations omitted). Under the enablement requirement of Section 112, a specification must "teach those skilled in the art how to make and use the full scope of the claimed invention without 'undue experimentation.'" *Streck, Inc. v. Res.*

*& Diagnostic Sys., Inc.*, 665 F.3d 1269, 1288 (Fed. Cir. 2012) (citations omitted). "The more a party claims, the broader the monopoly it demands, the more it must enable." *Amgen Inc. v. Sanofi*, 598 U.S. 594, 613 (2023).

A patent claim is also invalid for indefiniteness under Section 112 if its language, when "read in light of the specification . . . and the prosecution history, fail[s] to inform, with reasonable certainty, those skilled in the art about the scope of the invention." *Nautilus, Inc. v. Biosig Instruments, Inc.*, 572 U.S. 898, 901 (2014) (quotation marks omitted). Claim terms may also be indefinite "if a term does not have proper antecedent basis where such basis is not otherwise present by implication or the meaning is not reasonably ascertainable." *Halliburton Energy Servs., Inc. v. M-I LLC*, 514 F.3d 1244, 1249 (Fed. Cir. 2008) (citations omitted).

**A. Lack of Written Description**

At least under USAA's infringement theories, the Asserted Patents' specifications do not describe the full scope of its claims as required by 35 U.S.C. § 112, first paragraph, and the asserted patent claims are therefore invalid. For example, the following limitations of the Asserted Claims of the Asserted Patents fail to meet the written description requirement:

- "**the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping**" ('310 patent claims 4, 7, 13, '095 patent claim(s) 5, 13, 22), "**the plurality of monitoring criteria being based on magnetic ink character recognition (MICR) line detection and one or both of corner detection and edge detection**" ('310 patent claim 12, '095 patent claim 21): These limitations lack written description because, although the specification generically lists several possible monitoring criteria, the specification lacks sufficient "blaze marks" to show that USAA had possession of the claimed particular combinations of monitoring criteria, much less how to combine them

into a single system. *Fujikawa v. Wattanasin*, 93 F.3d 1559, 1571 (1996); *see also Univ. of Minn. v. Gilead Scis., Inc.*, 61 F.4th 1350, 1357 (2023).

**B. Enablement**

USAA's patents provide less detailed teachings than the prior art cited herein regarding how to enable the Asserted Claims. Accordingly, at least under USAA's infringement theories, and to the extent that USAA alleges that the prior art cited herein does not render the Asserted Claims obvious, the patent specifications would not have enabled a person of ordinary skill in the art to practice the claimed inventions without undue experimentation—and thus the asserted claims are invalid for lack of enablement. For example, the following limitations of the Asserted Claims of the Asserted Patents fail to meet the enablement requirement:

- "**monitoring criterion**" / "**monitoring criteria**" ('310 Patent all claims, '095 patent claims 1-5, 9-13, 18-22, 26-29): To the extent that USAA alleges limitations reciting generic or particular monitoring criteria, or particular combinations thereof are not obvious because it would have been allegedly challenging or unpredictable to implement such monitoring criteria in a mobile device for capturing an image of a check, that using such criteria to automatically capture an image would allegedly not necessarily improve a manual capture system, or that using such criteria would allegedly not produce an image sufficient for use for remote deposit, these limitations are also not enabled because the specifications of the '310 and '095 patents do not provide any further guidance as to how to implement such monitoring criteria, and instead simply refer to known techniques. *See, e.g.*, '310 patent at 7:66-67 ("Any known technique(s) for edge detection and/or corner detection may be used."); *id.* at

8:59-62 ("Any known MICR line detection technique(s) may be used by the camera 207, the mobile device 106, and/or the financial institution."); *id.* at 9:47-10:19 ("A histogram is a well known graph and may be used to display where all of the brightness levels contained in an image are found …. Any known technique for generating a histogram for an image (such as a grayscale image of the image 230) may be used."); *id.* at 10:54-56 ("Any technique for sending the digital image to the financial institution may be used.").

- "**optical character recognition is performed on the check [in the digital image], a magnetic ink character recognition (MICR) line on the check [in the digital image] is recognized, and an amount payable on the check [in the digital image] is validated**" ('310 patent claims 1, 2, 5, 6, 8, 9, 10, 11, 14, 15; '095 patent clams 1, 9, 18, 26): To the extent that USAA alleges that this limitation is not obvious because it would have been allegedly challenging or unpredictable to read information from an image of a check captured by a mobile device, this limitation is also not enabled because the specifications of the '310 and '095 patents do not provide more guidance than the prior art as to how to capture a better image or purport to improve techniques for reading information therefrom, and instead simply refer to "known image processing software or application(s)." *See, e.g.*, '310 patent at 4:56-59.
- "**the digital image is a modified version of the captured video frame**" ('095 patent claims 7, 24): To the extent that USAA alleges that this limitation is not obvious because it would have been allegedly challenging or unpredictable to modify an image of a check captured by a mobile device, for example by

converting the image to comply with ANSI X9.37 DSTU—2003 specifications, this limitation is also not enabled because the specifications of the ’310 and ’095 patents do not provide any further guidance as to how to modify mobile device images so as to comply with that standard. *See, e.g.*, ’310 patent, 12:36-47 (“In this example, the image transmitted to the server 322 may be in any format, such as JPEG or TIFF, insofar as the server software has the ability to convert that image into a Check 21 compliant format. Alternately, the bank may output an X9.37 file to the clearing system. The Check Clearing for the 21st Century Act (or Check 21 Act) is a United States federal law that allows the recipient of a paper check to create a digital version, thereby eliminating the need for further handling of the physical document. The Check 21 standard for electronic exchange is defined in the standard DSTU X9.37-2003 (“X9.37”). It is a binary interchange format.”).

- “**downloaded app**” / “**app for download**” / “**software abstraction layer**” (’310 patent claims 1, 2, 5, 6, 8, 9, 10, 11, 14, 15; ’095 patent clams 1, 9, 18, 26): To the extent that USAA alleges that this limitation is not obvious because it would have been allegedly challenging or unpredictable to provide a downloaded application that accesses a mobile device camera or one that does so using software abstraction layers (e.g., APIs) to do so, this limitation is also not enabled because the specifications of the ’310 and ’095 patents do not provide more guidance than on how to implement such features, and instead simply refer to prior art mobile device operating systems. *See, e.g.*, ’310 patent at 11:26-41.

## C. Indefiniteness

At least under USAA's infringement theories, the Asserted Claims do not "inform, with reasonable certainty, those skilled in the art about the scope of the invention," and are therefore invalid as indefinite. *Nautilus, Inc.*, 134 S. Ct. at 2124 (quotation marks omitted). For example, the following limitations of the Asserted Claims of the Asserted Patents are invalid as indefinite:

- "**wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising the monitoring, the determining, the providing, and the confirming**" ('310 Patent claim 8): This limitation is indefinite because the term "providing" finds antecedent basis in two prior limitations, rendering it unclear which "providing" step is included in the "additional operations." *See, e.g.*, '310 patent, claim 8 ("providing instructions to the user for capturing an image of the check"), ("providing feedback advising the user that the check has been captured").

## VII. SUBJECT MATTER INELIGIBILITY

Pursuant to the Court's Order Regarding Subject Matter Eligibility Contentions, Regions states that it has moved to dismiss all Counts of the Second Amended Complaint on the basis that the Asserted Patents are invalid pursuant to 35 U.S.C. § 101. Regions hereby incorporates its briefing in support of that motion into these contentions. Dkts. 14, 26, 46. The ineligibility of USAA's patents has also been briefed extensively in connection with PNC Bank's appeals in a parallel litigation involving USAA's mobile check deposit patents. *United Servs. Auto. Ass'n v. PNC Bank N.A.*, Nos. 2023-1639, -1866, 2025-1276, -1341 (Fed. Cir.); *United Servs. Auto. Ass'n v. PNC Bank N.A.*, Nos. 2023-1778, 2025-1277 (Fed. Cir.). Regions hereby incorporates the briefing submitted by PNC Bank in those appeals.

On June 12, 2025, the Federal Circuit issued two decisions invalidating at total of four USAA mobile check deposit patents under 35 U.S.C. § 101. *United Servs. Auto. Ass'n v. PNC Bank N.A.*, Nos. 2023-1639, -1866, 2025-1276, -1341, 2025 WL 1659297 (Fed. Cir. June 12, 2025) ("*PNC I*"); *United Servs. Auto. Ass'n v. PNC Bank N.A.*, Nos. 2023-1778, 2025-1277, 2025 WL 1662737 (Fed. Cir. June 12, 2025) ("*PNC II*").

Under *Alice* step one, the Federal Circuit held that USAA's patents were directed to the patent-ineligible "abstract idea of depositing a check using a handheld mobile device." *PNC II*, 2025 WL 1659297, at *3; *PNC I*, 2025 WL 1662737, at *2 ("the abstract idea of depositing a check using a mobile device"). The Federal Circuit rejected USAA's assertion that its patents claimed any technological improvement and explained that USAA's patent claims were instead "drafted in a result-oriented fashion without the requisite specificity needed to provide a non-abstract technological solution." *PNC II*, 2025 WL 1659297, at *4; *see also PNC I*, 2025 WL 1662737, at *3 ("There is no elaboration or specificity on how steps like 'checking for errors' or 'monitoring lighting' are performed—the patents just disclose that these steps happen and discuss them in a results-oriented manner.").

At *Alice* step two, the Federal Circuit held that USAA's patent claims merely recited the "computer-mediated implementation of routine or conventional activity [that] is not enough to provide an inventive concept." *PNC II*, 2025 WL 1659297, at *5; *see also PNC I*, 2025 WL 1662737, at *3. The court rejected USAA's argument that the ability to use existing consumer devices for check deposit provided an inventive concept. On the contrary, the Federal Circuit found that USAA's patents simply disclose and claim the use of existing technologies in a conventional way, which does not supply any inventive concept. *PNC II*, 2025 WL 1659297, at

*5; *PNC I*, 2025 WL 1662737, at *3 ("We do not find an inventive concept in the claims—only the implementation of routine activities using generic devices.").

As explained in Region's pending motion to dismiss, the Federal Circuit's *PNC* decisions are dispositive of the subject matter ineligibility of the two asserted patents in this litigation. There is significant overlap between the claims of USAA's patents that were invalidated under § 101 in *PNC*. For example, USAA's patent claims that were invalidated under § 101 in *PNC* included limitations mirroring the downloaded app, alignment guide, transmission, check processing, and bank account updating limitations of USAA's patents asserted here. The only additional limitations included in the patents asserted in this litigation relate to the use of a "monitoring criterion" to automatically capture check images. But those elements do not provide any patent eligible subject matter; on the contrary, those claim limitations are drafted using results-oriented language that fails to provide any non-abstract, technological solution. And like the other USAA patents that the Federal Circuit already invalidated, the claims of the asserted patents do not contain any inventive concept and instead simply recite generic computer components performing their ordinary functions and arranged in their usual manner. Indeed, USAA's own patent specification goes out of its way to emphasize that the alleged invention of the asserted patents may be practiced using entirely conventional technologies in their ordinary way. *See, e.g.*, '310 patent, 3:60-63 ("The mobile device 106 may be a mobile phone (also known as a wireless phone or a cellular phone), a personal data assistant (PDA), or any handheld computing device, for example."); *id.* at 4:56-5:6 (check images may be processed using "any known image processing software or other application(s)" and "any combination of systems and subsystems such as electronic devices including, but not limited to, computers, servers, databases, or the like"); *id.* at 7:66-67 ("Any known technique(s) for edge detection and/or

corner detection may be used."); *id.* at 8:59-62 ("Any known MICR line detection technique(s) may be used by the camera 207, the mobile device 106, and/or the financial institution."); *id.* at 9:47-10:19 ("A histogram is a well known graph and may be used to display where all of the brightness levels contained in an image are found …. Any known technique for generating a histogram for an image (such as a grayscale image of the image 230) may be used."); *id.* at 10:54-56 ("Any technique for sending the digital image to the financial institution may be used.").

Regions further provides the following disclosure:

**A. The '310 Patent**

Regions contends that each of the asserted claims of the '310 Patent is not patent eligible under 35 U.S.C. § 101 and *Alice Corp. v. CLS Bank Int'l*, 573 U.S. 208 (2014). Under *Alice* step one, all asserted claims of the '310 Patent are directed to the patent-ineligible abstract idea of verifying the quality of a check image for deposit. *See PNC II*, 2025 WL 1659297, at *3-4; *PNC I*, 2025 WL 1662737, at *2-3.

Monitoring an image and comparing it to "monitoring criterion" or an "alignment guide" is merely collecting and analyzing information. Automatically capturing the check image when it satisfies the criterion simply states the functional result of the analysis. Courts have repeatedly held that patent claims involving technologies similar to those at issue here are directed to an abstract idea when they recite nothing more than using generic computers to gather or analyze data. *See Mobile Acuity Ltd. v. Blippar Ltd.*, 110 F.4th 1280, 1293 (Fed. Cir. 2024) ("As we have by now frequently held, claims reciting generalized steps of collecting, analyzing, and presenting information, using nothing other than the conventional operations of generic computer components, are directed to abstract ideas."); *see also PNC II*, 2025 WL 1659297, at *4 ("[T]he claim recites routine data collection and analysis steps that have been traditionally performed by

banks and people depositing checks—namely reviewing checks, recognizing relevant data, checking for errors, and storing the resultant data. … The addition of a handheld mobile device to carry out these routine steps does not make the claim any less abstract."); *Electric Power Grp., LLC v. Alstrom S.A.*, 830 F.3d 1350, 1353-1354 (Fed. Cir. 2016); *Bozeman Fin. LLC v. Fed. Reserve Bank of Atl.*, 955 F.3d 971, 979 (Fed. Cir. 2020) ("recording or extracting data from physical documents, such as paper checks, is not alone sufficient to render claims not abstract"); *Content Extraction & Transmission LLC v. Wells Fargo Bank, Nat'l Ass'n*, 776 F.3d 1343, 1347-49 (Fed. Cir. 2014); *Solutran, Inc. v. Elavon, Inc.*, 931 F.3d 1161, 1166 (Fed. Cir. 2019) (claims abstract when reciting "[a] method for processing paper checks" involving "electronically receiving a data file" containing "an amount of a transaction associated with MICR information for each paper check" and "crediting an account for the merchant").

Reinforcing their abstract nature, all the asserted claims of the '310 Patent use purely result-oriented language, reciting only generic steps for capturing and validating check images without specifying how the claimed results are achieved. *See Apple, Inc. v. Ameranth, Inc.*, 842 F.3d 1229, 1241 (Fed. Cir. 2016) (finding claims abstract because "they do not claim a particular way of programming or designing the software" but "instead merely claim the resulting systems," and are thus "directed to certain functionality" rather than "a specific improvement" to computers); *see also Realtime Data LLC v. Array Networks Inc.*, Nos. 21-2251, -2291, 2023 WL 4924814, at *9 (Fed. Cir. Aug. 2, 2023) (claims that "do not explain" how the idea is implemented are abstract); *ChargePoint, Inc. v. SemaConnect, Inc.*, 920 F.3d 759, 769-70 (Fed. Cir. 2019) (claim directed to a "result-oriented" abstract idea is patent ineligible); *University of Fla. Research Found., Inc. v. General Electric Co.*, 916 F.3d 1363, 1368 (Fed. Cir. 2019) (data-conversion claims ineligible where patent failed to explain "*how* the drivers do the conversion,"

and because "[t]he mere *function* of converting is not a 'specific improvement to the way computers operate'" (emphases in original)).

The asserted claims of the '310 Patent also contain no inventive concept under *Alice* step two. The claims recite the use of generic computer components performing their ordinary functions and arranged in a traditional manner. For example, the '310 Patent does not purport to have invented mobile phones with cameras or the image processing software to monitor and automatically capture images. Instead, the '310 Patent's written description repeatedly emphasizes the conventionality of the underlying technology. *See* '310 Patent, 3:60-63 ("The mobile device 106 may be a mobile phone (also known as a wireless phone or a cellular phone), a personal data assistant (PDA), or any handheld computing device, for example."); *id.* at 4:56-5:6 (check images may be processed using "any known image processing software or other application(s)" and "any combination of systems and subsystems such as electronic devices including, but not limited to, computers, servers, databases, or the like"); *id.* at 7:66-67 ("Any known technique(s) for edge detection and/or corner detection may be used."); *id.* at 8:59-62 ("Any known MICR line detection technique(s) may be used by the camera 207, the mobile device 106, and/or the financial institution."); *id.* at 9:47-10:19 ("A histogram is a well known graph and may be used to display where all of the brightness levels contained in an image are found …. Any known technique for generating a histogram for an image (such as a grayscale image of the image 230) may be used."); *id.* at 10:54-56 ("Any technique for sending the digital image to the financial institution may be used."). Such conventional computer components performing their usual functions and arranged in their ordinary manner do not provide any inventive concept. *See, e.g.*, *Intellectual Ventures I LLC v. Capital One Fin. Corp.*, 850 F.3d 1322, 1341 (Fed. Cir. 2017) ("[T]he claims recite both a generic computer element—a

processor—and a series of generic computer 'components' that merely restate their individual functions—*i.e.*, organizing, mapping, identifying, defining, and modifying. … That is simply not enough under step two."); *Intellectual Ventures I LLC v. Symantec Corp.*, 838 F.3d 1307, 1315 (Fed. Cir. 2016) (claims "'in which each step does no more than require a generic computer to perform generic computer functions' do not make an abstract idea patent-eligible" under *Alice* step two) (quoting *Alice*, 573 U.S. at 225)); *Alice*, 573 U.S. at 225 ("[T]he use of a computer to obtain data, adjust account balances, and issue automated instructions" consists only of "'well-understood, routine, conventional activit[ies]' previously known to the industry."). The recited steps of the '310 Patent claims, obtaining or "monitoring" images, selecting or "capturing" an image that meets a criterion, and providing that image to a depository, are all part of the abstract idea of verifying a check image's quality before capturing or submitting it and lack an inventive concept. *See, e.g.*, *PNC II*, 2025 WL 1659297, at *4 ("USAA alleges that the claim read as a whole, considering the ordered combination of elements, contains an inventive concept because it solves the technological problem of accurate detection and extraction of information from digital images of checks using general purpose mobile devices. We disagree. The claim recites nothing more than routine image capture, OCR, and data processing steps—all of which were well-known and routine. … [T]here is no technological improvement—though the claim recites a system that makes the remote check deposit process easier and more convenient for bank customers, there is no fundamental change to how any of the technology functions, because it is all operating in a conventional way."); *Capital One*, 850 F.3d at 1341 (claims that "merely describe the functions of the abstract idea itself, without particularity," are "not enough under step two").

Further details regarding Regions' contentions that the asserted claims of the '310 Patent do not qualify as patent-eligible subject matter are contained in Exhibit C1. Specifically, pursuant to section (a)(1) of the Court's Order Regarding Subject Matter Eligibility Contentions, Exhibit C1contains a chart identifying each exception to eligibility to which each asserted claim is directed and the factual and legal basis therefor. For avoidance of doubt, and as set forth above and in Exhibit C1, each asserted claim of the '310 Patent is directed to an abstract idea and therefore is not directed to patent eligible subject matter. Exhibit C1 also identifies claim 1 as representative of the other asserted claims for purposes of patent eligibility. In addition, pursuant to section (a)(2)(b) and (a)(3) of the Court's Order Regarding Subject Matter Eligibility Contentions, Exhibit C1 provides: (i) a description of how each element of each asserted claim, both individually and in combination with other elements of that claim, was well understood, routine, and/or conventional in the relevant industry at the relevant time and (ii) an identification of the factual or legal basis for how the asserted claims are otherwise ineligible for patent protection.

**B. The '095 Patent**

Regions contends that each of the asserted claims of the '095 Patent is not patent eligible under 35 U.S.C. § 101 and *Alice Corp. v. CLS Bank Int'l*, 573 U.S. 208 (2014). Under *Alice* step one, all asserted claims of the '095 Patent are directed to the patent-ineligible abstract idea of verifying the quality of a check image for deposit. Monitoring an image and comparing it to "monitoring criterion" or an "alignment guide" is merely collecting and analyzing information. Automatically capturing the check image when it satisfies the criterion simply states the functional result of the analysis. Courts have repeatedly held that patent claims involving technologies similar to those at issue here are directed to an abstract idea when they recite nothing more than using generic computers to gather or analyze data. *See Mobile Acuity Ltd. v.*

*Blippar Ltd.*, 110 F.4th 1280, 1293 (Fed. Cir. 2024) ("As we have by now frequently held, claims reciting generalized steps of collecting, analyzing, and presenting information, using nothing other than the conventional operations of generic computer components, are directed to abstract ideas."); *see also PNC II*, 2025 WL 1659297, at *4 (Fed. Cir. June 12, 2025) ("[T]he claim recites routine data collection and analysis steps that have been traditionally performed by banks and people depositing checks—namely reviewing checks, recognizing relevant data, checking for errors, and storing the resultant data. … The addition of a handheld mobile device to carry out these routine steps does not make the claim any less abstract."); *Electric Power Grp., LLC v. Alstrom S.A.*, 830 F.3d 1350, 1353-1354 (Fed. Cir. 2016); *Bozeman Fin. LLC v. Fed. Reserve Bank of Atl*., 955 F.3d 971, 979 (Fed. Cir. 2020) ("recording or extracting data from physical documents, such as paper checks, is not alone sufficient to render claims not abstract"); *Content Extraction & Transmission LLC v. Wells Fargo Bank, Nat'l Ass'n*, 776 F.3d 1343, 1347-49 (Fed. Cir. 2014); *Solutran, Inc. v. Elavon, Inc.*, 931 F.3d 1161, 1166 (Fed. Cir. 2019) (claims abstract when reciting "[a] method for processing paper checks" involving "electronically receiving a data file" containing "an amount of a transaction associated with MICR information for each paper check" and "crediting an account for the merchant").

Reinforcing their abstract nature, all the asserted claims of the '095 Patent use purely result-oriented language, reciting only generic steps for capturing and validating check images without specifying how the claimed results are achieved. *See Apple, Inc. v. Ameranth, Inc.*, 842 F.3d 1229, 1241 (Fed. Cir. 2016) (finding claims abstract because "they do not claim a particular way of programming or designing the software" but "instead merely claim the resulting systems," and are thus "directed to certain functionality" rather than "a specific improvement" to computers); *see also Realtime Data LLC v. Array Networks Inc.*, Nos. 21-2251, -2291, 2023 WL

4924814, at *9 (Fed. Cir. Aug. 2, 2023) (claims that "do not explain" how the idea is implemented are abstract); *ChargePoint, Inc. v. SemaConnect, Inc.*, 920 F.3d 759, 769-70 (Fed. Cir. 2019) (claim directed to a "result-oriented" abstract idea is patent ineligible); *University of Fla. Research Found., Inc. v. General Electric Co.*, 916 F.3d 1363, 1368 (Fed. Cir. 2019) (data-conversion claims ineligible where patent failed to explain "*how* the drivers do the conversion," and because "[t]he mere *function* of converting is not a 'specific improvement to the way computers operate'" (emphases in original)).

The asserted claims of the '095 Patent also contain no inventive concept under *Alice* step two. The claims recite the use of generic computer components performing their ordinary functions and arranged in a traditional manner. For example, the '095 Patent does not purport to have invented mobile phones with cameras or the image processing software to monitor and automatically capture images. Instead, the '095 Patent's written description repeatedly emphasizes the conventionality of the underlying technology. *See* '095 Patent, 3:60-63 ("The mobile device 106 may be a mobile phone (also known as a wireless phone or a cellular phone), a personal data assistant (PDA), or any handheld computing device, for example."); *id.* at 4:56-5:6 (check images may be processed using "any known image processing software or other application(s)" and "any combination of systems and subsystems such as electronic devices including, but not limited to, computers, servers, databases, or the like"); *id.* at 7:66-67 ("Any known technique(s) for edge detection and/or corner detection may be used."); *id.* at 8:59-62 ("Any known MICR line detection technique(s) may be used by the camera 207, the mobile device 106, and/or the financial institution."); *id.* at 9:47-10:19 ("A histogram is a well known graph and may be used to display where all of the brightness levels contained in an image are found …. Any known technique for generating a histogram for an image (such as a grayscale

image of the image 230) may be used."); *id.* at 10:54-56 ("Any technique for sending the digital image to the financial institution may be used."). Such conventional computer components performing their usual functions and arranged in their ordinary manner do not provide any inventive concept. *See, e.g.*, *Intellectual Ventures I LLC v. Capital One Fin. Corp.*, 850 F.3d 1322, 1341 (Fed. Cir. 2017) ("[T]he claims recite both a generic computer element—a processor—and a series of generic computer 'components' that merely restate their individual functions—*i.e.*, organizing, mapping, identifying, defining, and modifying. … That is simply not enough under step two."); *Intellectual Ventures I LLC v. Symantec Corp.*, 838 F.3d 1307, 1315 (Fed. Cir. 2016) (claims "'in which each step does no more than require a generic computer to perform generic computer functions' do not make an abstract idea patent-eligible" under *Alice* step two) (quoting *Alice*, 573 U.S. at 225)); *Alice*, 573 U.S. at 225 ("[T]he use of a computer to obtain data, adjust account balances, and issue automated instructions" consists only of "'well-understood, routine, conventional activit[ies]' previously known to the industry."). The recited steps of the '095 Patent claims, obtaining or "monitoring" images, selecting or "capturing" an image that meets a criterion, and providing that image to a depository, are all part of the abstract idea of verifying a check image's quality before capturing or submitting it and lack an inventive concept. *See, e.g.*, *PNC II*, 2025 WL 1659297, at *4 ("USAA alleges that the claim read as a whole, considering the ordered combination of elements, contains an inventive concept because it solves the technological problem of accurate detection and extraction of information from digital images of checks using general purpose mobile devices. We disagree. The claim recites nothing more than routine image capture, OCR, and data processing steps—all of which were well-known and routine. … [T]here is no technological improvement—though the claim recites a system that makes the remote check deposit process easier and more convenient for bank

customers, there is no fundamental change to how any of the technology functions, because it is all operating in a conventional way."); *Capital One*, 850 F.3d at 1341 (claims that "merely describe the functions of the abstract idea itself, without particularity," are "not enough under step two").

Further details regarding Regions' contentions that the asserted claims of the '095 Patent do not qualify as patent-eligible subject matter are contained in Exhibit C2. Specifically, pursuant to section (a)(1) of the Court's Order Regarding Subject Matter Eligibility Contentions, Exhibit C2 contains a chart identifying each exception to eligibility to which each asserted claim is directed and the factual and legal basis therefor. For avoidance of doubt, and as set forth above and in Exhibit C2, each asserted claim of the '095 Patent is directed to an abstract idea and therefore is not directed to patent eligible subject matter. Exhibit C2 also identifies claim 1 as representative of the other asserted claims for purposes of patent eligibility. In addition, pursuant to section (a)(2)(b) and (a)(3) of the Court's Order Regarding Subject Matter Eligibility Contentions, Exhibit C2 provides: (i) a description of how each element of each asserted claim, both individually and in combination with other elements of that claim, was well understood, routine, and/or conventional in the relevant industry at the relevant time and (ii) an identification of the factual or legal basis for how the asserted claims are otherwise ineligible for patent protection.

Dated: July 9, 2025

Respectfully submitted,

*/s/ Andrew J. Danford*

Melisssa R. Smith
Texas State Bar No. 24001351
GILLAM & SMITH LLP
303 S. Washington Ave.
Marshall, TX 75670
(903) 934-8450
melissa@gillamsmithlaw.com

Gregory H. Lantier
R. Gregory Israelsen
WILMER CUTLER PICKERING HALE & DORR LLP
2100 Pennsylvania Avenue NW
Washington DC 20037
(202) 663-6000
gregory.lantier@wilmerhale.com
greg.israelsen@wilmerhale.com

Andrew J. Danford (*pro hac vice*)
Michaela P. Sewall (*pro hac vice*)
Nina Garcia (*pro hac vice*)
WILMER CUTLER PICKERING HALE & DORR LLP
60 State Street
Boston, MA 02109
andrew.danford@wilmerhale.com
michaela.sewall@wilmerhale.com
nina.garcia@wilmerhale.com

*Attorneys for Regions Bank*

**Exhibit A3 - Invalidity of 12,159,310 in view of Byrne**

US 2006/0249567 A1 ("Byrne") anticipates and/or renders obvious (alone or in combination with other references), at least under USAA's apparent infringement theories, all asserted claims of U.S. Patent No. 12,159,310 ("the '310 patent"). Byrne was filed on February 9, 2006 and published on November 9, 2006 and is therefore prior art to the '310 patent.

Regions has charted the portions of the reference that correspond to each element, including at least under USAA's apparent interpretation of the claims as reflected in Regions Complaint and Infringement Contentions. Where the chart below states that the reference "discloses," "teaches," or "suggests" a claim limitation, such disclosure, teaching, or suggestion may be express, implied, or inherent. The cited evidence is merely illustrative, and Regions reserves the right to cite alternative or additional evidence.

No part of this exemplary chart construes, or is intended to construe, the specification, file history, or claims of the asserted patents. In addition, nothing stated in this chart shall be treated as an admission or suggestion that Regions agrees with USAA regarding either the scope of any of the asserted claims or Regions' apparent interpretation of the scope of the asserted claims, or that Regions' accused products meet any limitations of the claims.

To the extent that USAA contends that Byrne does not disclose one or more limitations of the asserted claims, it would have been obvious to combine the teachings of Byrne with (1) the knowledge of one of ordinary skill in the art, and/or (2) the teachings of any of the prior art set forth in Regions' other invalidity charts and cover pleading, to show all the limitations of the asserted claims, including for the reasons set forth in Regions' Invalidity Contention Cover Pleading, which are incorporated herein.

The cited disclosure further shows that it would have been obvious to combine Byrne with the prior art set forth in Regions' other invalidity charts and cover pleading, for example by showing similarities with the other prior art and/or elements that could be improved by combination, replacement, and/or modification in view of the other prior art.

Because USAA has yet to identify any limitation of the asserted claims that it contends is not fully disclosed by Byrne, either alone or in combination with other prior art cited by Regions, Regions expressly reserves the right to rebut any such contention, including by identifying additional § 103 combinations, if any such contention is made by USAA.

| **Claims** | **Prior Art** |
|---|---|
| 1[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 249 of 1772

| Claims | Prior Art |
| --- | --- |
| | using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 250 of 1772

| Claims | Prior Art |
| --- | --- |
| | examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 251 of 1772

| Claims | Prior Art |
| --- | --- |
| | version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 252 of 1772

| Claims | Prior Art |
| --- | --- |
| | download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 253 of 1772

| Claims | Prior Art |
| --- | --- |
| | provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 254 of 1772

| Claims | Prior Art |
| --- | --- |
| | provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation; |

| Claims | Prior Art |
| --- | --- |
| | single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 256 of 1772

| Claims | Prior Art |
| --- | --- |
| | appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 257 of 1772

| Claims | Prior Art |
| --- | --- |
| | with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 259 of 1772

| Claims | Prior Art |
| --- | --- |
| | The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses |

| Claims | Prior Art |
| --- | --- |
| | JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 261 of 1772

| Claims | Prior Art |
| --- | --- |
| | posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 262 of 1772

| Claims | Prior Art |
| --- | --- |
| | ActiveX components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 263 of 1772

| Claims | Prior Art |
| --- | --- |
| | and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, |

| Claims | Prior Art |
|---|---|
| | coupon, or document.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1a] providing an app for download to a mobile device of a | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory |

| Claims | Prior Art |
| --- | --- |
| user, the user's mobile device including a camera and a memory storing a software abstraction layer for controlling the camera, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations comprising: | storing a software abstraction layer for controlling the camera, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and |

| Claims | Prior Art |
| --- | --- |
| | Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, |

| Claims | Prior Art |
| --- | --- |
| | U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and |

| Claims | Prior Art |
| --- | --- |
| | storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 269 of 1772

| Claims | Prior Art |
| --- | --- |
| | provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any |

| Claims | Prior Art |
| --- | --- |
| | type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0044]<br>Three of the components: PC, scanner, and device drivers can be packaged by any major computer manufacturer. The application software provider can either make special arrangements with the PC provider to include the software or the provider can use other means to distribute the software. The enterprise application is the responsibility of the enterprise.<br><br>Byrne at [0045]<br>Application software can be packaged in two basic forms. The first form is normally distributed by media, such as CD-ROM, or electronically downloaded. Once in the hands of the user, the application software is installed on the machine where it stays resident on disk and is loaded into memory when used. The software industry refers this type of application as a fat application, i.e. stays on machine. Application software, installed in this manner, also normally has access to the resources such as scanners, disk drives, and other applications available on the machine within the authorization scope of the user. Additionally, software maintenance, changes or new features for fat applications require the software provider to distribute the software and for the user to install the software. If the maintenance is faulty, additional roll-back steps or software maintenance is required. Because not all users can be expected to install software at the same time, software providers are forced into a position to support multiple, often back-leveled versions of software.<br><br>Byrne at [0046]<br>The second form is distributed with each invocation of the application software through the ubiquitous |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 271 of 1772

| Claims | Prior Art |
| --- | --- |
| | browser interface available on almost every modern personal computer. The software industry refers to this type of application as a thin client. No software is installed on the disk. The application software is the same version for all users because the plug-in is downloaded with each invocation of the application software, i.e. automatic maintenance. The plug-in does not automatically have the same level of access to the resources that the user has. This security feature allows the user to give permission to the resource being requested, e.g. scanner, not all of the resources available to the user. The combined benefits of security with ease and speed of software maintenance makes the thin-client approach a more beneficial approach to both the customer and the enterprise.<br><br>Byrne at [0047]<br>One embodiment of the invention can be described with reference to FIG. 1, a schematic flow diagram showing a high level flow of a desktop deposit according to the invention. From a client means, for example, a personal computer type computer (PC), a client/user authenticates with a desktop deposit application 102. In this example, the server related software is referred to as Wells Fargo's Commercial Electronic Offices$^{SM}$ Portal (CEO Portal or just CEO) and the desktop deposit application is referred to as C21 App. After authentication, C21 App is launched and the appropriate desktop deposit plug-in(s) is downloaded onto the client PC.<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 272 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 273 of 1772

| Claims | Prior Art |
| --- | --- |
| | provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0059]<br>In this particular example of one embodiment of the invention, an ActiveX component (VC++) is used for interacting with the device API, referred to herein as Ranger API, for scanning the checks. The Envoy client is used for retrieving the entitlements. The Envoy client uses T3 protocol, which is proprietary to WebLogic Application Server. DD Application Server uses HTTPS/Post for interfacing with AIM. A Wells Fargo standard, WFXML, is followed for making the request/response to AIM service. DD Application Server posts messages in X9.37 format to ICING MQSeries queue. DD Application Server uses standard JDBC implementation to interact with Work In Progress (local) Database. For retrieving Account Name needs to interact with internal system using Xstream Interface. For RTN validation an external system is used. It should be appreciated that the implementation described hereinabove is for illustrative purposes and is not meant to be limiting.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 274 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX |

| Claims | Prior Art |
| --- | --- |
| | component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction |

| Claims | Prior Art |
| --- | --- |
| | layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0142]–[0144]<br>Automated Updates.<br>The CLASP component 406 is designed as a signed ActiveX control. The control is deployed as a cab, i.e. Windows cabinet, file. This allows the control to be downloaded automatically whenever a new version of the control is posted to the CEO web server.<br>When the CEO application page references a new control version, the user's browser 408 automatically downloads or prompts for download of the control depending upon the browser security settings.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0149]–[0150]<br>Browser and OS.<br>By way of example, in one embodiment of the invention, the CLASP component is designed to work with the IE 5.01 (SP2) or IE 6 browsers. The operating systems supported are Windows XP and Windows 2000. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 277 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 278 of 1772

| Claims | Prior Art |
| --- | --- |
| | for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 279 of 1772

| Claims | Prior Art |
| --- | --- |
| | immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary |

| Claims | Prior Art |
| --- | --- |
| | information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 8<br>8. The computer implemented method of claim 1, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 10<br>10. The computer implemented method of claim 1, further comprising the step of: |

| Claims | Prior Art |
| --- | --- |
| | responsive to a new version of said one or more client components posted to said enterprise server component, said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, |

| Claims | Prior Art |
| --- | --- |
| | if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 283 of 1772

| Claims | Prior Art |
|---|---|
| | device types.<br><br>Byrne at claim 25<br>25. The system of claim 16, further comprising:<br>responsive to a new version of said one or more client components posted to said enterprise server component, means for said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitation 1[pre].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1b] providing instructions to the user for capturing an image of a check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of a check." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates |

| Claims | Prior Art |
| --- | --- |
| | to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing |

| Claims | Prior Art |
| --- | --- |
| | with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital |

| Claims | Prior Art |
| --- | --- |
| | signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed |

| Claims | Prior Art |
| --- | --- |
| | machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors |

| Claims | Prior Art |
| --- | --- |
| | which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application). |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included |

| Claims | Prior Art |
| --- | --- |
| | with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an |

| Claims | Prior Art |
| --- | --- |
| | argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of |

| Claims | Prior Art |
| --- | --- |
| | an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 294 of 1772

| Claims | Prior Art |
| --- | --- |
| | components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application; |

| Claims | Prior Art |
|---|---|
| | from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>*See also* limitation 1[pre].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [1c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively |

| Claims | Prior Art |
| --- | --- |
| | examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 298 of 1772

| Claims | Prior Art |
| --- | --- |
| | goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0044]<br>Three of the components: PC, scanner, and device drivers can be packaged by any major computer manufacturer. The application software provider can either make special arrangements with the PC provider to include the software or the provider can use other means to distribute the software. The enterprise application is the responsibility of the enterprise.<br><br>Byrne at [0045]<br>Application software can be packaged in two basic forms. The first form is normally distributed by media, such as CD-ROM, or electronically downloaded. Once in the hands of the user, the application software is installed on the machine where it stays resident on disk and is loaded into memory when used. The software industry refers this type of application as a fat application, i.e. stays on machine. Application software, installed in this manner, also normally has access to the resources such as scanners, disk drives, and other applications available on the machine within the authorization scope of the user. Additionally, software maintenance, changes or new features for fat applications require the software provider to distribute the software and for the user to install the software. If the maintenance is faulty, additional roll-back steps or software maintenance is required. Because not all users can be expected to install software at the same time, software providers are forced into a position to support multiple, often back-leveled versions of software.<br><br>Byrne at [0046]<br>The second form is distributed with each invocation of the application software through the ubiquitous browser interface available on almost every modern personal computer. The software industry refers to this type of application as a thin client. No software is installed on the disk. The application software is the same version for all users because the plug-in is downloaded with each invocation of the application software, i.e. automatic maintenance. The plug-in does not automatically have the same level of access to |

| Claims | Prior Art |
| --- | --- |
| | the resources that the user has. This security feature allows the user to give permission to the resource being requested, e.g. scanner, not all of the resources available to the user. The combined benefits of security with ease and speed of software maintenance makes the thin-client approach a more beneficial approach to both the customer and the enterprise.<br><br>Byrne at [0047]<br>One embodiment of the invention can be described with reference to FIG. 1, a schematic flow diagram showing a high level flow of a desktop deposit according to the invention. From a client means, for example, a personal computer type computer (PC), a client/user authenticates with a desktop deposit application 102. In this example, the server related software is referred to as Wells Fargo's Commercial Electronic Offices$^{SM}$ Portal (CEO Portal or just CEO) and the desktop deposit application is referred to as C21 App. After authentication, C21 App is launched and the appropriate desktop deposit plug-in(s) is downloaded onto the client PC.<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work |

| Claims | Prior Art |
| --- | --- |
| | with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0059]<br>In this particular example of one embodiment of the invention, an ActiveX component (VC++) is used for interacting with the device API, referred to herein as Ranger API, for scanning the checks. The Envoy client is used for retrieving the entitlements. The Envoy client uses T3 protocol, which is proprietary to |

| Claims | Prior Art |
| --- | --- |
| | WebLogic Application Server. DD Application Server uses HTTPS/Post for interfacing with AIM. A Wells Fargo standard, WFXML, is followed for making the request/response to AIM service. DD Application Server posts messages in X9.37 format to ICING MQSeries queue. DD Application Server uses standard JDBC implementation to interact with Work In Progress (local) Database. For retrieving Account Name needs to interact with internal system using Xstream Interface. For RTN validation an external system is used. It should be appreciated that the implementation described hereinabove is for illustrative purposes and is not meant to be limiting.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 303 of 1772

| Claims | Prior Art |
| --- | --- |
| | ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API. |

| Claims | Prior Art |
| --- | --- |
| | FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0142]–[0144]<br>Automated Updates.<br>The CLASP component 406 is designed as a signed ActiveX control. The control is deployed as a cab, i.e. Windows cabinet, file. This allows the control to be downloaded automatically whenever a new version of the control is posted to the CEO web server.<br>When the CEO application page references a new control version, the user's browser 408 automatically downloads or prompts for download of the control depending upon the browser security settings.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0149]–[0150]<br>Browser and OS.<br>By way of example, in one embodiment of the invention, the CLASP component is designed to work with the IE 5.01 (SP2) or IE 6 browsers. The operating systems supported are Windows XP and Windows 2000. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 306 of 1772

| Claims | Prior Art |
| --- | --- |
| | for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. |

| Claims | Prior Art |
| --- | --- |
| | 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 8<br>8. The computer implemented method of claim 1, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 10<br>10. The computer implemented method of claim 1, further comprising the step of:<br>responsive to a new version of said one or more client components posted to said enterprise server component, said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise |

| Claims | Prior Art |
| --- | --- |
| | server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 25<br>25. The system of claim 16, further comprising:<br>responsive to a new version of said one or more client components posted to said enterprise server component, means for said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component |

| Claims | Prior Art |
|---|---|
| | as an HTTP message.<br><br>*See also* limitations 1[pre], 1[a].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1d] overlaying an alignment guide on the live video of the check in the field of view of the camera; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, |

| Claims | Prior Art |
| --- | --- |
| | such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 313 of 1772

| Claims | Prior Art |
| --- | --- |
| | interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types. |

| Claims | Prior Art |
| --- | --- |
| | Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish |

| Claims | Prior Art |
| --- | --- |
| | scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 317 of 1772

| Claims | Prior Art |
| --- | --- |
| | ActiveX components.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 318 of 1772

| Claims | Prior Art |
| --- | --- |
| | for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that |

| Claims | Prior Art |
|---|---|
| | said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitation 1[pre], 1[a], 1[c].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1e] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical |

| Claims | Prior Art |
| --- | --- |
| | goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can |

| Claims | Prior Art |
| --- | --- |
| | detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility. |

| Claims | Prior Art |
| --- | --- |
| | The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses |

| Claims | Prior Art |
| --- | --- |
| | JMS to interact with ICING.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 325 of 1772

| Claims | Prior Art |
| --- | --- |
| | device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception |

| Claims | Prior Art |
| --- | --- |
| | conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[pre], 1[a], 1[d]. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 328 of 1772

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 329 of 1772

| Claims | Prior Art |
| --- | --- |
| | digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 330 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 331 of 1772

| Claims | Prior Art |
| --- | --- |
| | Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish |

| Claims | Prior Art |
| --- | --- |
| | scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 334 of 1772

| Claims | Prior Art |
| --- | --- |
| | ActiveX components.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 335 of 1772

| Claims | Prior Art |
| --- | --- |
| | for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that |

| Claims | Prior Art |
|---|---|
| | said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[pre], 1[a], 1[e].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience. |

| Claims | Prior Art |
| --- | --- |
| | It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and |

| Claims | Prior Art |
| --- | --- |
| | appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility. |

| Claims | Prior Art |
| --- | --- |
| | The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses |

| Claims | Prior Art |
| --- | --- |
| | JMS to interact with ICING.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 342 of 1772

| Claims | Prior Art |
| --- | --- |
| | device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 343 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception |

| Claims | Prior Art |
| --- | --- |
| | conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[pre], 1[a], 1[e], 1[f]. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 345 of 1772

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1h] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known |

| Claims | Prior Art |
| --- | --- |
| | problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted |

| Claims | Prior Art |
| --- | --- |
| | between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 348 of 1772

| Claims | Prior Art |
| --- | --- |
| | Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0014]<br>It would be advantageous to improve operating efficiencies by providing faster access to money deposited in a customer's account and providing increased flexibility to invest and direct funds.<br><br>Byrne at [0015]<br>It would further be advantageous to provide a system and method that can handle reductions in float as the time to settle transactions between banks decreases.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 349 of 1772

| Claims | Prior Art |
| --- | --- |
| | of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks |

| Claims | Prior Art |
| --- | --- |
| | using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been |

| Claims | Prior Art |
| --- | --- |
| | electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. |

| Claims | Prior Art |
| --- | --- |
| | The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 353 of 1772

| Claims | Prior Art |
| --- | --- |
| | download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0088]<br>This file format is based on the ANSI X9.37 DSTU—2003 “Specifications for Electronic Exchange of Check and Image Data” draft standard with some modifications to accommodate the processing of credits, such as customer Direct Deposit Account (DDA) deposits and adjustments, and to augment sender source identification and file balancing controls. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0089]<br>To ensure proper processing, it is important that all deposits on a file are transmitted in sequential order, and, for example, deposit credit before associated checks, and that each credit and debit item is assigned a unique, i.e. within business day, Item Sequence Number (ISN).<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 355 of 1772

| Claims | Prior Art |
| --- | --- |
| | What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in |

| Claims | Prior Art |
| --- | --- |
| | JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs |

| Claims | Prior Art |
| --- | --- |
| | immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0185]<br>The CEO system and method then takes the incoming request and access the current session deposit bag 622. It takes the session ID and tags it to each item, having identified the item sequence number.<br><br>Byrne at [0186]<br>The CEO system and method checks for any additional validations to be performed 624.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 359 of 1772

| Claims | Prior Art |
| --- | --- |
| | said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or |

| Claims | Prior Art |
| --- | --- |
| | more different capturing device types.<br><br>Byrne at claim 11<br>11. The computer implemented method of claim 1, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 13<br>13. The computer implemented method of claim 2, further comprising the step of:<br>providing a deposit bag that serves as a holding area until a deposit is committed.<br>Byrne at claim 14<br>14. The computer implemented method of claim 1, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented.<br><br>Byrne at claim 15<br>15. The computer implemented method of claim 1, wherein any of said captured one or more objects is replaced by a file upload.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 361 of 1772

| Claims | Prior Art |
| --- | --- |
| | components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 26<br>The system of claim 16, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 28<br>28. The system of claim 17, further comprising:<br>means for providing a deposit bag that serves as a holding area until a deposit is committed. |

| Claims | Prior Art |
|---|---|
| | Byrne at claim 29<br>29. The system of claim 16, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented.<br><br>Byrne at claim 30<br>30. The system of claim 16, wherein any of said captured one or more objects is replaced by a file upload.<br><br>*See also* limitations 1[pre], 1[a], 1[g].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1i] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and |

| Claims | Prior Art |
| --- | --- |
| | present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image |

| Claims | Prior Art |
| --- | --- |
| | processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a |

| Claims | Prior Art |
| --- | --- |
| | paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies |

| Claims | Prior Art |
| --- | --- |
| | of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks |

| Claims | Prior Art |
| --- | --- |
| | using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been |

| Claims | Prior Art |
| --- | --- |
| | electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. |

| Claims | Prior Art |
| --- | --- |
| | The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user: |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 371 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. |

| Claims | Prior Art |
| --- | --- |
| | In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0167]<br>Deposit bag is a holding/receiving area until the customer commits the deposit. Items are not committed to ICING until the bag is closed;<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item; and<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through |

| Claims | Prior Art |
| --- | --- |
| | 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0185]<br>The CEO system and method then takes the incoming request and access the current session deposit bag 622. It takes the session ID and tags it to each item, having identified the item sequence number.<br><br>Byrne at [0186]<br>The CEO system and method checks for any additional validations to be performed 624.<br><br>Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, |

| Claims | Prior Art |
| --- | --- |
| | back image, and/or receipt information is sent.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary |

| Claims | Prior Art |
| --- | --- |
| | information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 4<br>4. The computer implemented method of claim 1, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 13<br>13. The computer implemented method of claim 2, further comprising the step of:<br>providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 376 of 1772

| Claims | Prior Art |
| --- | --- |
| | means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 19<br>19. The system of claim 16, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 28<br>28. The system of claim 17, further comprising:<br>means for providing a deposit bag that serves as a holding area until a deposit is committed.<br>*See also* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [1j] receiving the digital image of the check transmitted by the user's mobile device; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving the digital image of the check transmitted by the user's mobile device." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for |

| Claims | Prior Art |
| --- | --- |
| | Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of |

| Claims | Prior Art |
| --- | --- |
| | paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be |

| Claims | Prior Art |
| --- | --- |
| | processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 381 of 1772

| Claims | Prior Art |
| --- | --- |
| | provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any |

| Claims | Prior Art |
| --- | --- |
| | digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability |

| Claims | Prior Art |
| --- | --- |
| | to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to |

| Claims | Prior Art |
| --- | --- |
| | enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and |

| Claims | Prior Art |
| --- | --- |
| | What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses |

| Claims | Prior Art |
| --- | --- |
| | JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0185]<br>The CEO system and method then takes the incoming request and access the current session deposit bag 622. It takes the session ID and tags it to each item, having identified the item sequence number.<br><br>Byrne at [0186]<br>The CEO system and method checks for any additional validations to be performed 624.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional |

| Claims | Prior Art |
| --- | --- |
| | information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 11<br>11. The computer implemented method of claim 1, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 13<br>13. The computer implemented method of claim 2, further comprising the step of:<br>providing a deposit bag that serves as a holding area until a deposit is committed.<br>Byrne at claim 14<br>14. The computer implemented method of claim 1, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 15<br>15. The computer implemented method of claim 1, wherein any of said captured one or more objects is replaced by a file upload.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components. |

| Claims | Prior Art |
|---|---|
| | Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 26<br>The system of claim 16, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 28<br>28. The system of claim 17, further comprising:<br>means for providing a deposit bag that serves as a holding area until a deposit is committed.<br>Byrne at claim 29<br>29. The system of claim 16, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented.<br><br>Byrne at claim 30<br>30. The system of claim 16, wherein any of said captured one or more objects is replaced by a file upload.<br><br>*See also* limitations 1[h], 1[i]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1k] processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward." For example:<br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the |

| Claims | Prior Art |
| --- | --- |
| | customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and |

| Claims | Prior Art |
| --- | --- |
| | inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. |

| Claims | Prior Art |
| --- | --- |
| | O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and |

| Claims | Prior Art |
| --- | --- |
| | deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can |

| Claims | Prior Art |
| --- | --- |
| | perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For |

| Claims | Prior Art |
| --- | --- |
| | instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window. |

| Claims | Prior Art |
| --- | --- |
| | Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0120]–[0121]<br>Keep Alive Utility<br>This is a utility, which is useful when on a particular screen and background processes are taking some time for processing. It shows the user a screen displaying information about progress information.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 400 of 1772

| Claims | Prior Art |
| --- | --- |
| | invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0167]<br>Deposit bag is a holding/receiving area until the customer commits the deposit. Items are not committed to ICING until the bag is closed;<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item; and<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of |

| Claims | Prior Art |
| --- | --- |
| | an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0180]<br>receive optional image(s) based on a flag means.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0185]<br>The CEO system and method then takes the incoming request and access the current session deposit bag 622. It takes the session ID and tags it to each item, having identified the item sequence number.<br><br>Byrne at [0186]<br>The CEO system and method checks for any additional validations to be performed 624. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within |

| Claims | Prior Art |
| --- | --- |
| | scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 4<br>4. The computer implemented method of claim 1, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 13<br>13. The computer implemented method of claim 2, further comprising the step of:<br>providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 404 of 1772

| Claims | Prior Art |
| --- | --- |
| | from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 19<br>19. The system of claim 16, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 28<br>28. The system of claim 17, further comprising:<br>means for providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>*See also* limitation 1[i]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1l] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit, | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known |

| Claims | Prior Art |
| --- | --- |
| | problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted |

| Claims | Prior Art |
| --- | --- |
| | between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a |

| Claims | Prior Art |
| --- | --- |
| | Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0014]<br>It would be advantageous to improve operating efficiencies by providing faster access to money deposited in a customer's account and providing increased flexibility to invest and direct funds.<br><br>Byrne at [0015]<br>It would further be advantageous to provide a system and method that can handle reductions in float as the time to settle transactions between banks decreases.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 409 of 1772

| Claims | Prior Art |
| --- | --- |
| | of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks |

| Claims | Prior Art |
| --- | --- |
| | using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been |

| Claims | Prior Art |
| --- | --- |
| | electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. |

| Claims | Prior Art |
| --- | --- |
| | The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0088]<br>This file format is based on the ANSI X9.37 DSTU—2003 "Specifications for Electronic Exchange of Check and Image Data" draft standard with some modifications to accommodate the processing of credits, such as customer Direct Deposit Account (DDA) deposits and adjustments, and to augment sender source |

| Claims | Prior Art |
| --- | --- |
| | identification and file balancing controls.<br><br>Byrne at [0089]<br>To ensure proper processing, it is important that all deposits on a file are transmitted in sequential order, and, for example, deposit credit before associated checks, and that each credit and debit item is assigned a unique, i.e. within business day, Item Sequence Number (ISN).<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and |

| Claims | Prior Art |
| --- | --- |
| | package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit |

| Claims | Prior Art |
| --- | --- |
| | checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise |

| Claims | Prior Art |
| --- | --- |
| | backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application |

| Claims | Prior Art |
| --- | --- |
| | and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>*See also* limitations 1[h], 1[i], 1[k]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1m] wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations." For example:<br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks |

| Claims | Prior Art |
| --- | --- |
| | from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 420 of 1772

| Claims | Prior Art |
| --- | --- |
| | provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers |

| Claims | Prior Art |
| --- | --- |
| | provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0044]<br>Three of the components: PC, scanner, and device drivers can be packaged by any major computer manufacturer. The application software provider can either make special arrangements with the PC provider to include the software or the provider can use other means to distribute the software. The enterprise application is the responsibility of the enterprise.<br><br>Byrne at [0045]<br>Application software can be packaged in two basic forms. The first form is normally distributed by media, such as CD-ROM, or electronically downloaded. Once in the hands of the user, the application software is installed on the machine where it stays resident on disk and is loaded into memory when used. The software industry refers this type of application as a fat application, i.e. stays on machine. Application software, installed in this manner, also normally has access to the resources such as scanners, disk drives, and other applications available on the machine within the authorization scope of the user. Additionally, software maintenance, changes or new features for fat applications require the software provider to distribute the software and for the user to install the software. If the maintenance is faulty, additional roll-back steps or software maintenance is required. Because not all users can be expected to install software at the same time, software providers are forced into a position to support multiple, often back-leveled versions of software.<br><br>Byrne at [0046]<br>The second form is distributed with each invocation of the application software through the ubiquitous browser interface available on almost every modern personal computer. The software industry refers to this type of application as a thin client. No software is installed on the disk. The application software is the same version for all users because the plug-in is downloaded with each invocation of the application software, i.e. automatic maintenance. The plug-in does not automatically have the same level of access to the resources that the user has. This security feature allows the user to give permission to the resource being requested, e.g. scanner, not all of the resources available to the user. The combined benefits of security with ease and speed of software maintenance makes the thin-client approach a more beneficial |

| Claims | Prior Art |
| --- | --- |
| | approach to both the customer and the enterprise.<br><br>Byrne at [0047]<br>One embodiment of the invention can be described with reference to FIG. 1, a schematic flow diagram showing a high level flow of a desktop deposit according to the invention. From a client means, for example, a personal computer type computer (PC), a client/user authenticates with a desktop deposit application 102. In this example, the server related software is referred to as Wells Fargo's Commercial Electronic Offices[SM] Portal (CEO Portal or just CEO) and the desktop deposit application is referred to as C21 App. After authentication, C21 App is launched and the appropriate desktop deposit plug-in(s) is downloaded onto the client PC.<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability |

| Claims | Prior Art |
| --- | --- |
| | to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0059]<br>In this particular example of one embodiment of the invention, an ActiveX component (VC++) is used for interacting with the device API, referred to herein as Ranger API, for scanning the checks. The Envoy client is used for retrieving the entitlements. The Envoy client uses T3 protocol, which is proprietary to WebLogic Application Server. DD Application Server uses HTTPS/Post for interfacing with AIM. A Wells Fargo standard, WFXML, is followed for making the request/response to AIM service. DD |

| Claims | Prior Art |
| --- | --- |
| | Application Server posts messages in X9.37 format to ICING MQSeries queue. DD Application Server uses standard JDBC implementation to interact with Work In Progress (local) Database. For retrieving Account Name needs to interact with internal system using Xstream Interface. For RTN validation an external system is used. It should be appreciated that the implementation described hereinabove is for illustrative purposes and is not meant to be limiting.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this |

| Claims | Prior Art |
| --- | --- |
| | functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0142]–[0144]<br>Automated Updates.<br>The CLASP component 406 is designed as a signed ActiveX control. The control is deployed as a cab, i.e. Windows cabinet, file. This allows the control to be downloaded automatically whenever a new version of the control is posted to the CEO web server.<br>When the CEO application page references a new control version, the user's browser 408 automatically downloads or prompts for download of the control depending upon the browser security settings.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0149]–[0150]<br>Browser and OS.<br>By way of example, in one embodiment of the invention, the CLASP component is designed to work with the IE 5.01 (SP2) or IE 6 browsers. The operating systems supported are Windows XP and Windows 2000.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 427 of 1772

| Claims | Prior Art |
| --- | --- |
| | In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 428 of 1772

| Claims | Prior Art |
| --- | --- |
| | Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the |

| Claims | Prior Art |
| --- | --- |
| | Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or |

| Claims | Prior Art |
| --- | --- |
| | ActiveX components.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 8<br>8. The computer implemented method of claim 1, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 10<br>10. The computer implemented method of claim 1, further comprising the step of:<br>responsive to a new version of said one or more client components posted to said enterprise server component, said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 431 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 432 of 1772

| Claims | Prior Art |
| --- | --- |
| | device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 25<br>25. The system of claim 16, further comprising:<br>responsive to a new version of said one or more client components posted to said enterprise server component, means for said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[pre], 1[a]. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 433 of 1772

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1n] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known |

| Claims | Prior Art |
| --- | --- |
| | problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted |

| Claims | Prior Art |
| --- | --- |
| | between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 436 of 1772

| Claims | Prior Art |
| --- | --- |
| | Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and |

| Claims | Prior Art |
| --- | --- |
| | deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0044]<br>Three of the components: PC, scanner, and device drivers can be packaged by any major computer manufacturer. The application software provider can either make special arrangements with the PC provider to include the software or the provider can use other means to distribute the software. The enterprise application is the responsibility of the enterprise.<br><br>Byrne at [0045]<br>Application software can be packaged in two basic forms. The first form is normally distributed by media, such as CD-ROM, or electronically downloaded. Once in the hands of the user, the application software is installed on the machine where it stays resident on disk and is loaded into memory when used. The software industry refers this type of application as a fat application, i.e. stays on machine. Application software, installed in this manner, also normally has access to the resources such as scanners, disk drives, and other applications available on the machine within the authorization scope of the user. Additionally, software maintenance, changes or new features for fat applications require the software provider to |

| Claims | Prior Art |
| --- | --- |
| | distribute the software and for the user to install the software. If the maintenance is faulty, additional roll-back steps or software maintenance is required. Because not all users can be expected to install software at the same time, software providers are forced into a position to support multiple, often back-leveled versions of software.<br><br>Byrne at [0046]<br>The second form is distributed with each invocation of the application software through the ubiquitous browser interface available on almost every modern personal computer. The software industry refers to this type of application as a thin client. No software is installed on the disk. The application software is the same version for all users because the plug-in is downloaded with each invocation of the application software, i.e. automatic maintenance. The plug-in does not automatically have the same level of access to the resources that the user has. This security feature allows the user to give permission to the resource being requested, e.g. scanner, not all of the resources available to the user. The combined benefits of security with ease and speed of software maintenance makes the thin-client approach a more beneficial approach to both the customer and the enterprise.<br><br>Byrne at [0047]<br>One embodiment of the invention can be described with reference to FIG. 1, a schematic flow diagram showing a high level flow of a desktop deposit according to the invention. From a client means, for example, a personal computer type computer (PC), a client/user authenticates with a desktop deposit application 102. In this example, the server related software is referred to as Wells Fargo's Commercial Electronic Offices[SM] Portal (CEO Portal or just CEO) and the desktop deposit application is referred to as C21 App. After authentication, C21 App is launched and the appropriate desktop deposit plug-in(s) is downloaded onto the client PC.<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug- |

| Claims | Prior Art |
| --- | --- |
| | in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The |

| Claims | Prior Art |
| --- | --- |
| | uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0059]<br>In this particular example of one embodiment of the invention, an ActiveX component (VC++) is used for interacting with the device API, referred to herein as Ranger API, for scanning the checks. The Envoy client is used for retrieving the entitlements. The Envoy client uses T3 protocol, which is proprietary to WebLogic Application Server. DD Application Server uses HTTPS/Post for interfacing with AIM. A Wells Fargo standard, WFXML, is followed for making the request/response to AIM service. DD Application Server posts messages in X9.37 format to ICING MQSeries queue. DD Application Server uses standard JDBC implementation to interact with Work In Progress (local) Database. For retrieving Account Name needs to interact with internal system using Xstream Interface. For RTN validation an external system is used. It should be appreciated that the implementation described hereinabove is for illustrative purposes and is not meant to be limiting.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., |

| Claims | Prior Art |
| --- | --- |
| | and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), |

| Claims | Prior Art |
| --- | --- |
| | based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and |

| Claims | Prior Art |
| --- | --- |
| | What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0142]–[0144]<br>Automated Updates.<br>The CLASP component 406 is designed as a signed ActiveX control. The control is deployed as a cab, i.e. Windows cabinet, file. This allows the control to be downloaded automatically whenever a new version of the control is posted to the CEO web server.<br>When the CEO application page references a new control version, the user's browser 408 automatically downloads or prompts for download of the control depending upon the browser security settings. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0149]–[0150]<br>Browser and OS.<br>By way of example, in one embodiment of the invention, the CLASP component is designed to work with the IE 5.01 (SP2) or IE 6 browsers. The operating systems supported are Windows XP and Windows 2000.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 446 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses |

| Claims | Prior Art |
| --- | --- |
| | a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 448 of 1772

| Claims | Prior Art |
| --- | --- |
| | from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 8<br>8. The computer implemented method of claim 1, wherein said one or more client components provide an |

| Claims | Prior Art |
| --- | --- |
| | interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 10<br>10. The computer implemented method of claim 1, further comprising the step of:<br>responsive to a new version of said one or more client components posted to said enterprise server component, said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 450 of 1772

| Claims | Prior Art |
| --- | --- |
| | means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX |

| Claims | Prior Art |
|---|---|
| | controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 25<br>25. The system of claim 16, further comprising:<br>responsive to a new version of said one or more client components posted to said enterprise server component, means for said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[pre], 1[a],1[e], 1[f], 1[m].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 452 of 1772

| Claims | Prior Art |
|---|---|
| [1o] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively |

| Claims | Prior Art |
| --- | --- |
| | examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 454 of 1772

| Claims | Prior Art |
| --- | --- |
| | goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0044]<br>Three of the components: PC, scanner, and device drivers can be packaged by any major computer manufacturer. The application software provider can either make special arrangements with the PC provider to include the software or the provider can use other means to distribute the software. The enterprise application is the responsibility of the enterprise.<br><br>Byrne at [0045]<br>Application software can be packaged in two basic forms. The first form is normally distributed by media, such as CD-ROM, or electronically downloaded. Once in the hands of the user, the application software is installed on the machine where it stays resident on disk and is loaded into memory when used. The software industry refers this type of application as a fat application, i.e. stays on machine. Application software, installed in this manner, also normally has access to the resources such as scanners, disk drives, and other applications available on the machine within the authorization scope of the user. Additionally, software maintenance, changes or new features for fat applications require the software provider to distribute the software and for the user to install the software. If the maintenance is faulty, additional roll-back steps or software maintenance is required. Because not all users can be expected to install software at the same time, software providers are forced into a position to support multiple, often back-leveled versions of software.<br><br>Byrne at [0046]<br>The second form is distributed with each invocation of the application software through the ubiquitous browser interface available on almost every modern personal computer. The software industry refers to this type of application as a thin client. No software is installed on the disk. The application software is the same version for all users because the plug-in is downloaded with each invocation of the application software, i.e. automatic maintenance. The plug-in does not automatically have the same level of access to |

| Claims | Prior Art |
| --- | --- |
| | the resources that the user has. This security feature allows the user to give permission to the resource being requested, e.g. scanner, not all of the resources available to the user. The combined benefits of security with ease and speed of software maintenance makes the thin-client approach a more beneficial approach to both the customer and the enterprise.<br><br>Byrne at [0047]<br>One embodiment of the invention can be described with reference to FIG. 1, a schematic flow diagram showing a high level flow of a desktop deposit according to the invention. From a client means, for example, a personal computer type computer (PC), a client/user authenticates with a desktop deposit application 102. In this example, the server related software is referred to as Wells Fargo's Commercial Electronic Offices[SM] Portal (CEO Portal or just CEO) and the desktop deposit application is referred to as C21 App. After authentication, C21 App is launched and the appropriate desktop deposit plug-in(s) is downloaded onto the client PC.<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work |

| Claims | Prior Art |
| --- | --- |
| | with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0059]<br>In this particular example of one embodiment of the invention, an ActiveX component (VC++) is used for interacting with the device API, referred to herein as Ranger API, for scanning the checks. The Envoy client is used for retrieving the entitlements. The Envoy client uses T3 protocol, which is proprietary to |

| Claims | Prior Art |
| --- | --- |
| | WebLogic Application Server. DD Application Server uses HTTPS/Post for interfacing with AIM. A Wells Fargo standard, WFXML, is followed for making the request/response to AIM service. DD Application Server posts messages in X9.37 format to ICING MQSeries queue. DD Application Server uses standard JDBC implementation to interact with Work In Progress (local) Database. For retrieving Account Name needs to interact with internal system using Xstream Interface. For RTN validation an external system is used. It should be appreciated that the implementation described hereinabove is for illustrative purposes and is not meant to be limiting.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and |

| Claims | Prior Art |
| --- | --- |
| | ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 460 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0142]–[0144]<br>Automated Updates.<br>The CLASP component 406 is designed as a signed ActiveX control. The control is deployed as a cab, i.e. Windows cabinet, file. This allows the control to be downloaded automatically whenever a new version of the control is posted to the CEO web server.<br>When the CEO application page references a new control version, the user's browser 408 automatically downloads or prompts for download of the control depending upon the browser security settings.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0149]–[0150]<br>Browser and OS.<br>By way of example, in one embodiment of the invention, the CLASP component is designed to work with the IE 5.01 (SP2) or IE 6 browsers. The operating systems supported are Windows XP and Windows 2000.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device |

| Claims | Prior Art |
| --- | --- |
| | One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit |

| Claims | Prior Art |
| --- | --- |
| | checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 464 of 1772

| Claims | Prior Art |
| --- | --- |
| | from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 8<br>8. The computer implemented method of claim 1, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 10<br>10. The computer implemented method of claim 1, further comprising the step of:<br>responsive to a new version of said one or more client components posted to said enterprise server component, said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application; |

| Claims | Prior Art |
| --- | --- |
| | from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 467 of 1772

| Claims | Prior Art |
|---|---|
| | Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 25<br>25. The system of claim 16, further comprising:<br>responsive to a new version of said one or more client components posted to said enterprise server component, means for said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[pre], 1[a], 1[b], 1[e], 1[f], 1[h], 1[m], 1[n].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 2[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br><br>*See* limitation 1[pre].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [2a] providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer for controlling the camera, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations comprising: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer for controlling the camera, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2b] providing instructions to the user for capturing an image of a check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[b]<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [2d] overlaying an alignment guide on the live video of the check in the field of view of the camera; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2e] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [2h] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2i] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2j] receiving the digital image of the check transmitted by the user's mobile device; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving the digital image of the check transmitted by the user's mobile device." For example:<br>*See* limitation 1[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2k] processing the digital image of the check including performing optical character recognition on the check in the digital image and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile |

| Claims | Prior Art |
|---|---|
| verifying that the digital image of the check is acceptable for mobile check deposit, wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward; and | device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward." For example:<br><br>*See* limitation 1[k].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2l] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit, | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br><br>*See* limitation 1[l].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2m] wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example: |

| Claims | Prior Art |
| --- | --- |
| spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks |

| Claims | Prior Art |
| --- | --- |
| | from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a |

| Claims | Prior Art |
| --- | --- |
| | scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience. |

| Claims | Prior Art |
| --- | --- |
| | It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following: |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 478 of 1772

| Claims | Prior Art |
| --- | --- |
| | scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING |

| Claims | Prior Art |
| --- | --- |
| | Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One |

| Claims | Prior Art |
| --- | --- |
| | embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item; and<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit |

| Claims | Prior Art |
| --- | --- |
| | checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. |

| Claims | Prior Art |
| --- | --- |
| | 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 4<br>4. The computer implemented method of claim 1, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception |

| Claims | Prior Art |
|---|---|
| | conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 19<br>19. The system of claim 16, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>*See also* limitation 1[e].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 3. The method of claim 2, wherein the plurality of monitoring criteria includes a criterion based on magnetic ink character recognition (MICR) line detection. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 2, wherein the plurality of monitoring criteria includes a criterion based on magnetic ink character recognition (MICR) line detection." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real- |

| Claims | Prior Art |
| --- | --- |
| | time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively |

| Claims | Prior Art |
| --- | --- |
| | voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital |

| Claims | Prior Art |
| --- | --- |
| | signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 488 of 1772

| Claims | Prior Art |
| --- | --- |
| | machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors |

| Claims | Prior Art |
| --- | --- |
| | which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application). |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included |

| Claims | Prior Art |
| --- | --- |
| | with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an |

| Claims | Prior Art |
| --- | --- |
| | argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item; and<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit |

| Claims | Prior Art |
| --- | --- |
| | checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. |

| Claims | Prior Art |
| --- | --- |
| | 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 4<br>4. The computer implemented method of claim 1, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception |

| Claims | Prior Art |
|---|---|
| | conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 19<br>19. The system of claim 16, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>*See also* limitation 2[m].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 4. The method of claim 2, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 2, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping." For example:<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have |

| Claims | Prior Art |
| --- | --- |
| | printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of |

| Claims | Prior Art |
| --- | --- |
| | depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 500 of 1772

| Claims | Prior Art |
| --- | --- |
| | compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an |

| Claims | Prior Art |
| --- | --- |
| | acknowledgment for each scanned item; and<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and |

| Claims | Prior Art |
| --- | --- |
| | compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 4<br>4. The computer implemented method of claim 1, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 503 of 1772

| Claims | Prior Art |
|---|---|
| | means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 19<br>19. The system of claim 16, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>*See also* limitation 2[m].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 5[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br><br>*See* limitation 1[pre].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5a] monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example: |

| Claims | Prior Art |
|---|---|
| device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | *See* limitations 1[a], 1[c], 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5b] providing instructions to the user for capturing an image of the check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5c] generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5d] overlaying an alignment guide on the live video of the check in the field of view of the camera; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [5e] using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured." For example:<br>*See* limitations 1[e]–1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5h] processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitations 1[k], 1[g]. |

| Claims | Prior Art |
|---|---|
| frame or being based on the captured video frame; | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5i] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5j] updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5k] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example: |

| Claims | Prior Art |
|---|---|
| that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | *See* limitation 1[n].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5l] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 1[o].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 6[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6a] monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 5[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | |
| [6b] providing instructions to the user for capturing an image of the check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6c] generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6d] overlaying an alignment guide on the live video of the check in the field of view of the camera; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6e] using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured." For example:<br>*See* limitation 1[e]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6h] processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitation 5[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6i] confirming that the mobile check deposit can go forward after | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in |

| Claims | Prior Art |
|---|---|
| optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; and | the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 5[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6j] updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit, | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6k] wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* limitation 2[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| 7. The method of claim 6, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 6, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping." For example:<br>*See* claim 4.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 8[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8a] monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 5[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [8b] providing instructions to the user for capturing an image of the check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8c] generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8d] overlaying an alignment guide on the live video of the check in the field of view of the camera; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8e] using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured." For example:<br>*See* limitation 5[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [8f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8h] processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitation 5[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8i] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 5[i]. |

| Claims | Prior Art |
|---|---|
| on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; and | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8j] updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit, | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 5[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8k] wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising the monitoring, the determining, the providing, and the confirming. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising the monitoring, the determining, the providing, and the confirming." For example:<br>*See* limitations 1[a], 1[b], 1[f]–1[h], 5[a], 5[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 9[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | *See* limitation 1[pre].<br>To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example: |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9a] monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 5[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9b] providing instructions to the user for capturing an image of the check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9c] generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[c]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9d] overlaying an alignment guide on the live video of the check in the field of view of the camera; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9e] using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured." For example:<br>*See* limitations 5[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g]. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 517 of 1772

| Claims | Prior Art |
|---|---|
| providing feedback advising the user that the check has been captured; | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9h] processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitation 5[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9i] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 5[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9j] updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 5[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 518 of 1772

| Claims | Prior Art |
|---|---|
| confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit, | |
| [9k] wherein the confirming step is performed before the submitting step and the user's mobile device receives an indication of a result of the processing the digital image of the check before the confirming step. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the confirming step is performed before the submitting step and the user's mobile device receives an indication of a result of the processing the digital image of the check before the confirming step." For example:<br>*See* limitation 5[i], 1[i], 1[k].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 10[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10a] providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| by the user's mobile device, is configured to cause the user's mobile device to perform operations comprising: | |
| [10b] providing instructions to the user for capturing an image of a check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10d] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10e] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and the | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and the automatically capturing the video frame." For example:<br>*See* limitation 1[f]. |

| Claims | Prior Art |
|---|---|
| automatically capturing the video frame; | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10f] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10g] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10h] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [10i] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward, | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward." For example:<br>*See* limitation 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10j] wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations." For example:<br>*See* limitation 1[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10k] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 1[n].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10l] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 1[o]. |

| Claims | Prior Art |
|---|---|
| criterion, providing feedback to the user. | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 11[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11a] providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations comprising: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 10[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11b] providing instructions to the user for capturing an image of a check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[b]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11d] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11e] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11f] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example: |

| Claims | Prior Art |
|---|---|
| one monitoring criterion, providing feedback advising the user that the check has been captured; | *See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11g] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11h] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 10[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11i] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward." For example:<br>*See* limitation 10[i]. |

| Claims | Prior Art |
|---|---|
| and the confirming that the mobile check deposit can go forward, | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11j] wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* limitation 2[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 12. The method of claim 11, wherein the plurality of monitoring criteria being based on magnetic ink character recognition (MICR) line detection and one or both of corner detection and edge detection. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 11, wherein the plurality of monitoring criteria being based on magnetic ink character recognition (MICR) line detection and one or both of corner detection and edge detection." For example:<br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments |

| Claims | Prior Art |
| --- | --- |
| | system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. |

| Claims | Prior Art |
| --- | --- |
| | Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a |

| Claims | Prior Art |
| --- | --- |
| | non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century |

| Claims | Prior Art |
| --- | --- |
| | Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can |

| Claims | Prior Art |
| --- | --- |
| | be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, |

| Claims | Prior Art |
| --- | --- |
| | for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 533 of 1772

| Claims | Prior Art |
| --- | --- |
| | following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 534 of 1772

| Claims | Prior Art |
| --- | --- |
| | form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item; and<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the |

| Claims | Prior Art |
| --- | --- |
| | MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 4<br>4. The computer implemented method of claim 1, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for |

| Claims | Prior Art |
| --- | --- |
| | deposit is a bank check, coupon, or document.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more |

| Claims | Prior Art |
|---|---|
| | captured objects is one or more scanned images.<br><br>Byrne at claim 19<br>19. The system of claim 16, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>*See* claims 3, 4.<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 13. The method of claim 11, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 11, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping." For example:<br><br>*See* claim 4.<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 14[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br><br>*See* limitation 1[pre].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [14a] receiving an image of a check, wherein the image of the check is provided for mobile check deposit in a bank by a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer and a downloaded app, the downloaded app associated with the bank to control check deposit by controlling the camera through the software abstraction layer, and the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving an image of a check, wherein the image of the check is provided for mobile check deposit in a bank by a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer and a downloaded app, the downloaded app associated with the bank to control check deposit by controlling the camera through the software abstraction layer, and the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing |

| Claims | Prior Art |
| --- | --- |
| | image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a |

| Claims | Prior Art |
| --- | --- |
| | method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business |

| Claims | Prior Art |
| --- | --- |
| | goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0044]<br>Three of the components: PC, scanner, and device drivers can be packaged by any major computer manufacturer. The application software provider can either make special arrangements with the PC provider to include the software or the provider can use other means to distribute the software. The enterprise application is the responsibility of the enterprise.<br><br>Byrne at [0045]<br>Application software can be packaged in two basic forms. The first form is normally distributed by media, such as CD-ROM, or electronically downloaded. Once in the hands of the user, the application software is installed on the machine where it stays resident on disk and is loaded into memory when used. The software industry refers this type of application as a fat application, i.e. stays on machine. Application software, installed in this manner, also normally has access to the resources such as scanners, disk drives, and other applications available on the machine within the authorization scope of the user. Additionally, software maintenance, changes or new features for fat applications require the software provider to distribute the software and for the user to install the software. If the maintenance is faulty, additional roll-back steps or software maintenance is required. Because not all users can be expected to install software at the same time, software providers are forced into a position to support multiple, often back-leveled |

| Claims | Prior Art |
| --- | --- |
| | versions of software.<br><br>Byrne at [0046]<br>The second form is distributed with each invocation of the application software through the ubiquitous browser interface available on almost every modern personal computer. The software industry refers to this type of application as a thin client. No software is installed on the disk. The application software is the same version for all users because the plug-in is downloaded with each invocation of the application software, i.e. automatic maintenance. The plug-in does not automatically have the same level of access to the resources that the user has. This security feature allows the user to give permission to the resource being requested, e.g. scanner, not all of the resources available to the user. The combined benefits of security with ease and speed of software maintenance makes the thin-client approach a more beneficial approach to both the customer and the enterprise.<br><br>Byrne at [0047]<br>One embodiment of the invention can be described with reference to FIG. 1, a schematic flow diagram showing a high level flow of a desktop deposit according to the invention. From a client means, for example, a personal computer type computer (PC), a client/user authenticates with a desktop deposit application 102. In this example, the server related software is referred to as Wells Fargo's Commercial Electronic Offices[SM] Portal (CEO Portal or just CEO) and the desktop deposit application is referred to as C21 App. After authentication, C21 App is launched and the appropriate desktop deposit plug-in(s) is downloaded onto the client PC.<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is |

| Claims | Prior Art |
| --- | --- |
| | a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM |

| Claims | Prior Art |
| --- | --- |
| | is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0059]<br>In this particular example of one embodiment of the invention, an ActiveX component (VC++) is used for interacting with the device API, referred to herein as Ranger API, for scanning the checks. The Envoy client is used for retrieving the entitlements. The Envoy client uses T3 protocol, which is proprietary to WebLogic Application Server. DD Application Server uses HTTPS/Post for interfacing with AIM. A Wells Fargo standard, WFXML, is followed for making the request/response to AIM service. DD Application Server posts messages in X9.37 format to ICING MQSeries queue. DD Application Server uses standard JDBC implementation to interact with Work In Progress (local) Database. For retrieving Account Name needs to interact with internal system using Xstream Interface. For RTN validation an external system is used. It should be appreciated that the implementation described hereinabove is for illustrative purposes and is not meant to be limiting.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 548 of 1772

| Claims | Prior Art |
| --- | --- |
| | problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component |

| Claims | Prior Art |
| --- | --- |
| | The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0142]–[0144]<br>Automated Updates.<br>The CLASP component 406 is designed as a signed ActiveX control. The control is deployed as a cab, i.e. Windows cabinet, file. This allows the control to be downloaded automatically whenever a new version of the control is posted to the CEO web server.<br>When the CEO application page references a new control version, the user's browser 408 automatically downloads or prompts for download of the control depending upon the browser security settings.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0149]–[0150]<br>Browser and OS. |

| Claims | Prior Art |
| --- | --- |
| | By way of example, in one embodiment of the invention, the CLASP component is designed to work with the IE 5.01 (SP2) or IE 6 browsers. The operating systems supported are Windows XP and Windows 2000.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), |

| Claims | Prior Art |
| --- | --- |
| | invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form |

| Claims | Prior Art |
| --- | --- |
| | text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend |

| Claims | Prior Art |
| --- | --- |
| | post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 8<br>8. The computer implemented method of claim 1, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or |

| Claims | Prior Art |
| --- | --- |
| | more different capturing device types.<br><br>Byrne at claim 10<br>10. The computer implemented method of claim 1, further comprising the step of:<br>responsive to a new version of said one or more client components posted to said enterprise server component, said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message |

| Claims | Prior Art |
| --- | --- |
| | consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that |

| Claims | Prior Art |
|---|---|
| | said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 25<br>25. The system of claim 16, further comprising:<br>responsive to a new version of said one or more client components posted to said enterprise server component, means for said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See* limitations 1[a], 1[h], 1[j].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14b] providing instructions to the user for capturing the check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing the check." For example:<br><br>*See* limitation 1[b].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [14c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14d] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14e] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14f] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14g] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14h] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 10[1h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14i] performing image processing operations on the image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "performing image processing operations on the image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device." For example:<br>*See* limitations 1[h], 1[k].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 560 of 1772

| Claims | Prior Art |
|---|---|
| transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device; and | |
| [14j] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit, | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14k] wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations." For example:<br>*See* limitation 1[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14l] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 1[n]. |

| Claims | Prior Art |
|---|---|
| satisfy the at least one monitoring criterion; and | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14m] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 1[p].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 15[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15a] receiving an image of a check, wherein the image of the check is provided for mobile check deposit in a bank by a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer and a downloaded app, the downloaded app associated with the bank to control check deposit by controlling the camera through the software abstraction layer, and the | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving an image of a check, wherein the image of the check is provided for mobile check deposit in a bank by a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer and a downloaded app, the downloaded app associated with the bank to control check deposit by controlling the camera through the software abstraction layer, and the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 14[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | |
| [15b] providing instructions to the user for capturing the check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15d] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15e] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example: |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 563 of 1772

| Claims | Prior Art |
|---|---|
| automatically capturing the video frame; | *See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15f] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15g] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15h] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 10[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [15i] performing image processing operations on the image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "performing image processing operations on the image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device." For example:<br>*See* limitation 14[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15j] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit, | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 14[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15k] wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* limitation 2[m]. |

| Claims | Prior Art |
|---|---|
| spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 16. The method of claim 15, wherein the plurality of monitoring criteria includes a criterion based on magnetic ink character recognition (MICR) line detection. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 15, wherein the plurality of monitoring criteria includes a criterion based on magnetic ink character recognition (MICR) line detection." For example:<br>*See* claim 3.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

**Exhibit A6 - Invalidity of 12,159,310 in view of Garcia**

WO 2005/043857 ("Garcia") anticipates and/or renders obvious (alone or in combination with other references), at least under USAA's apparent infringement theories, all asserted claims of U.S. Patent No. 12,159,310 ("the '310 patent"). Garcia was filed on October 30, 2003 and published on May 12, 2005 and is therefore prior art to the '310 patent.

Regions has charted the portions of the reference that correspond to each element, including at least under USAA's apparent interpretation of the claims as reflected in Regions Complaint and Infringement Contentions. Where the chart below states that the reference "discloses," "teaches," or "suggests" a claim limitation, such disclosure, teaching, or suggestion may be express, implied, or inherent. The cited evidence is merely illustrative, and Regions reserves the right to cite alternative or additional evidence.

No part of this exemplary chart construes, or is intended to construe, the specification, file history, or claims of the asserted patents. In addition, nothing stated in this chart shall be treated as an admission or suggestion that Regions agrees with USAA regarding either the scope of any of the asserted claims or Regions' apparent interpretation of the scope of the asserted claims, or that Regions' accused products meet any limitations of the claims.

To the extent that USAA contends that Garcia does not disclose one or more limitations of the asserted claims, it would have been obvious to combine the teachings of Garcia with (1) the knowledge of one of ordinary skill in the art, and/or (2) the teachings of any of the prior art set forth in Regions' other invalidity charts and cover pleading, to show all the limitations of the asserted claims, including for the reasons set forth in Regions' Invalidity Contention Cover Pleading, which are incorporated herein.

The cited disclosure further shows that it would have been obvious to combine Garcia with the prior art set forth in Regions' other invalidity charts and cover pleading, for example by showing similarities with the other prior art and/or elements that could be improved by combination, replacement, and/or modification in view of the other prior art.

Because USAA has yet to identify any limitation of the asserted claims that it contends is not fully disclosed by Garcia, either alone or in combination with other prior art cited by Regions, Regions expressly reserves the right to rebut any such contention, including by identifying additional § 103 combinations, if any such contention is made by USAA.

| **Claims** | **Prior Art** |
|---|---|
| 1[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 567 of 1772

| Claims | Prior Art |
| --- | --- |
| | above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 568 of 1772

| Claims | Prior Art |
| --- | --- |
| | Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 569 of 1772

| Claims | Prior Art |
| --- | --- |
| | Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 570 of 1772

| Claims | Prior Art |
| --- | --- |
| | documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 571 of 1772

| Claims | Prior Art |
| --- | --- |
| | particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 572 of 1772

| Claims | Prior Art |
|---|---|
| | Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1a] providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer for controlling the camera, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations comprising: | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer for controlling the camera, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations." For example:<br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks. |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems |

| Claims | Prior Art |
|---|---|
| | nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 575 of 1772

| Claims | Prior Art |
| --- | --- |
| | capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention. |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 577 of 1772

| Claims | Prior Art |
| --- | --- |
| | original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 578 of 1772

| Claims | Prior Art |
|---|---|
| | sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitation 1[pre].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1b] providing instructions to the user for capturing an image of a check; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of a check." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending |

| Claims | Prior Art |
| --- | --- |
| | images via an Internet connection.<br><br>Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting |

| Claims | Prior Art |
| --- | --- |
| | digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all |

| Claims | Prior Art |
| --- | --- |
| | these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 582 of 1772

| Claims | Prior Art |
| --- | --- |
| | Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected. |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitation 1[pre]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the |

| Claims | Prior Art |
| --- | --- |
| | process and storage capacity of the documents.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution. |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 587 of 1772

| Claims | Prior Art |
| --- | --- |
| | documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, |

| Claims | Prior Art |
| --- | --- |
| | particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 589 of 1772

| Claims | Prior Art |
|---|---|
| | Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitations 1[pre], 1[a].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1d] overlaying an alignment guide on the live video of the check in the field of view of the camera; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 590 of 1772

| Claims | Prior Art |
| --- | --- |
| | such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 591 of 1772

| Claims | Prior Art |
| --- | --- |
| | mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 592 of 1772

| Claims | Prior Art |
| --- | --- |
| | Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 593 of 1772

| Claims | Prior Art |
| --- | --- |
| | c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 594 of 1772

| Claims | Prior Art |
|---|---|
| | mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitation 1[pre], 1[a], 1[c].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1e] monitoring the live video of the check in the field of view of | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example: |

| Claims | Prior Art |
| --- | --- |
| the camera with respect to at least one monitoring criterion; | Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via |

| Claims | Prior Art |
| --- | --- |
| | a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 597 of 1772

| Claims | Prior Art |
| --- | --- |
| | capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention. |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 599 of 1772

| Claims | Prior Art |
| --- | --- |
| | original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 600 of 1772

| Claims | Prior Art |
|---|---|
| | sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitations 1[pre], 1[a], 1[d].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending |

| Claims | Prior Art |
| --- | --- |
| | images via an Internet connection.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 602 of 1772

| Claims | Prior Art |
| --- | --- |
| | authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi- |

| Claims | Prior Art |
| --- | --- |
| | purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document. |

| Claims | Prior Art |
| --- | --- |
| | e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 605 of 1772

| Claims | Prior Art |
|---|---|
| | an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitations 1[pre], 1[a], 1[e].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 606 of 1772

| Claims | Prior Art |
| --- | --- |
| user that the check has been captured; | digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment. |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 608 of 1772

| Claims | Prior Art |
| --- | --- |
| | Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method. |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes |

| Claims | Prior Art |
| --- | --- |
| | necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 611 of 1772

| Claims | Prior Art |
|---|---|
| | camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitations 1[pre], 1[a], 1[e], 1[f].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1h] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending |

| Claims | Prior Art |
| --- | --- |
| | images via an Internet connection.<br><br>Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting |

| Claims | Prior Art |
| --- | --- |
| | digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all |

| Claims | Prior Art |
| --- | --- |
| | these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 615 of 1772

| Claims | Prior Art |
| --- | --- |
| | Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 11:7–19<br>For its part, the institution that renders the service must have certain electronic means of recognizing, verifying and processing the information sent by the user. These means consist basically of computer equipment connected to the Internet that has an optical character recognition (OCR) system, a code authentication system and means of transmitting data to the institution's document processing system. The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems).<br><br>Garcia at 11:20–12:3<br>The communication link between each user and the institution consists of a mobile telephone communications network, for example a GSM/GPRS network or any future technology with the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 616 of 1772

| Claims | Prior Art |
| --- | --- |
| | technological capabilities adapted to the present and future demands of the method of reference.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution. |

| Claims | Prior Art |
|---|---|
| | Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitations 1[pre], 1[a], 1[g].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1i] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks. |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems |

| Claims | Prior Art |
|---|---|
| | nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 620 of 1772

| Claims | Prior Art |
| --- | --- |
| | capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention. |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 11:7–19<br>For its part, the institution that renders the service must have certain electronic means of recognizing, |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 622 of 1772

| Claims | Prior Art |
| --- | --- |
| | verifying and processing the information sent by the user. These means consist basically of computer equipment connected to the Internet that has an optical character recognition (OCR) system, a code authentication system and means of transmitting data to the institution's document processing system. The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems).<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must |

| Claims | Prior Art |
|---|---|
| | accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1j] receiving the digital image of the check transmitted by the user's mobile device; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving the digital image of the check transmitted by the user's mobile device." For example:<br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and |

| Claims | Prior Art |
| --- | --- |
| | automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer. |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 626 of 1772

| Claims | Prior Art |
| --- | --- |
| | Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 627 of 1772

| Claims | Prior Art |
| --- | --- |
| | documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, |

| Claims | Prior Art |
| --- | --- |
| | particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 11:7–19<br>For its part, the institution that renders the service must have certain electronic means of recognizing, verifying and processing the information sent by the user. These means consist basically of computer equipment connected to the Internet that has an optical character recognition (OCR) system, a code authentication system and means of transmitting data to the institution's document processing system. The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems).<br><br>Garcia at 11:20–12:3<br>The communication link between each user and the institution consists of a mobile telephone communications network, for example a GSM/GPRS network or any future technology with the technological capabilities adapted to the present and future demands of the method of reference.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or |

| Claims | Prior Art |
| --- | --- |
| | steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitations 1[h], 1[i].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
| --- | --- |
| [1k] processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, |

| Claims | Prior Art |
| --- | --- |
| | because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 632 of 1772

| Claims | Prior Art |
| --- | --- |
| | b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 633 of 1772

| Claims | Prior Art |
| --- | --- |
| | and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI). |

| Claims | Prior Art |
| --- | --- |
| | d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 11:7–19<br>For its part, the institution that renders the service must have certain electronic means of recognizing, verifying and processing the information sent by the user. These means consist basically of computer equipment connected to the Internet that has an optical character recognition (OCR) system, a code authentication system and means of transmitting data to the institution's document processing system. The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems).<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of |

| Claims | Prior Art |
| --- | --- |
| | the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitation 1[i]. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 636 of 1772

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1l] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit, | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by |

| Claims | Prior Art |
| --- | --- |
| | the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 638 of 1772

| Claims | Prior Art |
| --- | --- |
| | applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 639 of 1772

| Claims | Prior Art |
| --- | --- |
| | communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 640 of 1772

| Claims | Prior Art |
| --- | --- |
| | b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 11:7–19<br>For its part, the institution that renders the service must have certain electronic means of recognizing, verifying and processing the information sent by the user. These means consist basically of computer equipment connected to the Internet that has an optical character recognition (OCR) system, a code authentication system and means of transmitting data to the institution's document processing system. The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems).<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set |

| Claims | Prior Art |
| --- | --- |
| | that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 642 of 1772

| Claims | Prior Art |
|---|---|
| | telephone via the mobile communications operator.<br><br>*See also* limitations 1[h], 1[i], 1[k].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1m] wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising: | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via |

| Claims | Prior Art |
| --- | --- |
| | a mobile telephone or similar portable equipment.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points. |

| Claims | Prior Art |
|---|---|
| | g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>*See also* limitations 1[pre], 1[a].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1n] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, |

| Claims | Prior Art |
| --- | --- |
| | such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial |

| Claims | Prior Art |
| --- | --- |
| | documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi- |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 647 of 1772

| Claims | Prior Art |
| --- | --- |
| | purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document. |

| Claims | Prior Art |
| --- | --- |
| | e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 649 of 1772

| Claims | Prior Art |
|---|---|
| | an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitations 1[pre], 1[a],1[e], 1[f], 1[m].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1o] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 650 of 1772

| Claims | Prior Art |
| --- | --- |
| | characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 651 of 1772

| Claims | Prior Art |
| --- | --- |
| | documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global |

| Claims | Prior Art |
| --- | --- |
| | positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device |

| Claims | Prior Art |
| --- | --- |
| | by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 654 of 1772

| Claims | Prior Art |
| --- | --- |
| | that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 655 of 1772

| Claims | Prior Art |
|---|---|
| | telephone via the mobile communications operator.<br><br>*See also* limitations 1[pre], 1[a], 1[b], 1[e], 1[f], 1[h], 1[m], 1[n].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 2[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2a] providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer for controlling the camera, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations comprising: | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer for controlling the camera, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [2b] providing instructions to the user for capturing an image of a check; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[b]<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2d] overlaying an alignment guide on the live video of the check in the field of view of the camera; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2e] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [2f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2h] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2i] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example: |

| Claims | Prior Art |
|---|---|
| digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; | *See* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2j] receiving the digital image of the check transmitted by the user's mobile device; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving the digital image of the check transmitted by the user's mobile device." For example:<br>*See* limitation 1[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2k] processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward." For example:<br>*See* limitation 1[k].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2l] updating, by a computer, a balance of an account at a financial institution of the user to reflect an | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting |

| Claims | Prior Art |
|---|---|
| amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit, | the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br><br>*See* limitation 1[l].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2m] wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending |

| Claims | Prior Art |
| --- | --- |
| | images via an Internet connection.<br><br>Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 661 of 1772

| Claims | Prior Art |
| --- | --- |
| | digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all |

| Claims | Prior Art |
| --- | --- |
| | these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 663 of 1772

| Claims | Prior Art |
| --- | --- |
| | Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 11:7–19<br>For its part, the institution that renders the service must have certain electronic means of recognizing, verifying and processing the information sent by the user. These means consist basically of computer equipment connected to the Internet that has an optical character recognition (OCR) system, a code authentication system and means of transmitting data to the institution's document processing system. The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems).<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 664 of 1772

| Claims | Prior Art |
| --- | --- |
| | aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 665 of 1772

| Claims | Prior Art |
|---|---|
| | Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitation 1[e].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 3. The method of claim 2, wherein the plurality of monitoring criteria includes a criterion based on magnetic ink character recognition (MICR) line detection. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 2, wherein the plurality of monitoring criteria includes a criterion based on magnetic ink character recognition (MICR) line detection." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection. |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations. |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile |

| Claims | Prior Art |
| --- | --- |
| | communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 669 of 1772

| Claims | Prior Art |
| --- | --- |
| | Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 11:7–19<br>For its part, the institution that renders the service must have certain electronic means of recognizing, verifying and processing the information sent by the user. These means consist basically of computer equipment connected to the Internet that has an optical character recognition (OCR) system, a code authentication system and means of transmitting data to the institution's document processing system. The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems).<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 670 of 1772

| Claims | Prior Art |
| --- | --- |
| | aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital |

| Claims | Prior Art |
|---|---|
| | camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See also* limitation 2[m].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 4. The method of claim 2, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 2, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending |

| Claims | Prior Art |
| --- | --- |
| | images via an Internet connection.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 673 of 1772

| Claims | Prior Art |
| --- | --- |
| | applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile |

| Claims | Prior Art |
| --- | --- |
| | communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 675 of 1772

| Claims | Prior Art |
| --- | --- |
| | b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 11:7–19<br>For its part, the institution that renders the service must have certain electronic means of recognizing, verifying and processing the information sent by the user. These means consist basically of computer equipment connected to the Internet that has an optical character recognition (OCR) system, a code authentication system and means of transmitting data to the institution's document processing system. The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems).<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set |

| Claims | Prior Art |
| --- | --- |
| | that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 677 of 1772

| Claims | Prior Art |
|---|---|
| | telephone via the mobile communications operator.<br><br>*See also* limitation 2[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 5[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5a] monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitations 1[a], 1[c], 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [5b] providing instructions to the user for capturing an image of the check; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5c] generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5d] overlaying an alignment guide on the live video of the check in the field of view of the camera; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5e] using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured." For example:<br>*See* limitations 1[e]–1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [5f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5h] processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitations 1[k], 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5i] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 1[i]. |

| Claims | Prior Art |
|---|---|
| on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5j] updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5k] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 1[n].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5l] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 1[o]. |

| Claims | Prior Art |
|---|---|
| criterion, providing feedback to the user. | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 6[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6a] monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 5[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6b] providing instructions to the user for capturing an image of the check; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of the check." For example:<br>*See* limitation 1[b]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6c] generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6d] overlaying an alignment guide on the live video of the check in the field of view of the camera; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6e] using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6h] processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitation 5[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6i] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 5[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [6j] updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit, | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6k] wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* limitation 2[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 7. The method of claim 6, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 6, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping." For example:<br>*See* claim 4.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| 8[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8a] monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 5[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8b] providing instructions to the user for capturing an image of the check; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [8c] generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8d] overlaying an alignment guide on the live video of the check in the field of view of the camera; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8e] using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured." For example:<br>*See* limitation 5[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [8g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8h] processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitation 5[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8i] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 5[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8j] updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is |

| Claims | Prior Art |
|---|---|
| associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit, | submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 5[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8k] wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising the monitoring, the determining, the providing, and the confirming. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising the monitoring, the determining, the providing, and the confirming." For example:<br>*See* limitations 1[a], 1[b], 1[f]–1[h], 5[a], 5[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 9[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | *See* limitation 1[pre].<br>To the extent the preamble is limiting, Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9a] monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the |

| Claims | Prior Art |
|---|---|
| device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 5[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9b] providing instructions to the user for capturing an image of the check; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9c] generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9d] overlaying an alignment guide on the live video of the check in the field of view of the camera; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9e] using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured." For example:<br>*See* limitations 5[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9h] processing a digital image of the check including performing optical character recognition and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing a digital image of the check including performing optical character recognition and verifying that the digital |

| Claims | Prior Art |
|---|---|
| verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame; | image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitation 5[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9i] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 5[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9j] updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit, | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 5[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [9k] wherein the confirming step is performed before the submitting step and the user's mobile device receives an indication of a result of the processing the digital image of the check before the confirming step. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the confirming step is performed before the submitting step and the user's mobile device receives an indication of a result of the processing the digital image of the check before the confirming step." For example:<br>*See* limitation 5[i], 1[i], 1[k].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 10[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10a] providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations comprising: | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [10b] providing instructions to the user for capturing an image of a check; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10d] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10e] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and the automatically capturing the video frame; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and the automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [10f] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10g] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10h] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10i] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting |

| Claims | Prior Art |
|---|---|
| the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward, | the check for mobile check deposit and the confirming that the mobile check deposit can go forward." For example:<br>*See* limitation 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10j] wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising: | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations." For example:<br>*See* limitation 1[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10k] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 1[n].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10l] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 1[o].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| 11[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11a] providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations comprising: | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 10[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11b] providing instructions to the user for capturing an image of a check; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [11c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11d] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11e] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11f] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11g] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11h] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 10[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11i] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward." For example:<br>*See* limitation 10[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 699 of 1772

| Claims | Prior Art |
|---|---|
| and the confirming that the mobile check deposit can go forward, | |
| [11j] wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* limitation 2[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 12. The method of claim 11, wherein the plurality of monitoring criteria being based on magnetic ink character recognition (MICR) line detection and one or both of corner detection and edge detection. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 11, wherein the plurality of monitoring criteria being based on magnetic ink character recognition (MICR) line detection and one or both of corner detection and edge detection." For example:<br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, |

| Claims | Prior Art |
| --- | --- |
| | using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of |

| Claims | Prior Art |
| --- | --- |
| | checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication |

| Claims | Prior Art |
| --- | --- |
| | codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi-purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device |

| Claims | Prior Art |
| --- | --- |
| | by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br><br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number.<br>f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 11:7–19<br>For its part, the institution that renders the service must have certain electronic means of recognizing, verifying and processing the information sent by the user. These means consist basically of computer equipment connected to the Internet that has an optical character recognition (OCR) system, a code authentication system and means of transmitting data to the institution's document processing system. The receiving institution makes available to the user a digital image of the document in its bank account and processes the document in question by the usual means (truncation systems). |

| Claims | Prior Art |
| --- | --- |
| | Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 705 of 1772

| Claims | Prior Art |
|---|---|
| | Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See* claims 3, 4.<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 13. The method of claim 11, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 11, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping." For example:<br><br>*See* claim 4.<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 14[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br><br>*See* limitation 1[pre].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 706 of 1772

| Claims | Prior Art |
| --- | --- |
| [14a] receiving an image of a check, wherein the image of the check is provided for mobile check deposit in a bank by a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer and a downloaded app, the downloaded app associated with the bank to control check deposit by controlling the camera through the software abstraction layer, and the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving an image of a check, wherein the image of the check is provided for mobile check deposit in a bank by a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer and a downloaded app, the downloaded app associated with the bank to control check deposit by controlling the camera through the software abstraction layer, and the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br><br>Garcia at Abstract<br>The invention relates to a method for the transmission, authentication and automatic processing of digitized documents in a single session, using a mobile device which is provided with the technological characteristics necessary for the aforementioned processes to be performed. The inventive method also comprises the recognition, verification and electronic processing of the information received by the receiving institution, which is incorporated automatically into the standard processing applied to the above-mentioned document(s). The method is particularly suitable for authenticating, transmitting and automatically processing financial and non-financial documents, such as bank checks.<br><br>Garcia at 1:2–3<br>Method for the transmission, authentication and automatic processing of digitized documents.<br><br>Garcia at 1:6–13<br>The present invention, as its title implies, relates to a method for processing and accepting documents, such as, for example, bank checks, with guarantee of authentication of the issuer and the document itself, using electronic media for its implementation through a mobile device capable of capturing and sending images via an Internet connection.<br><br>Garcia at 1:15–22<br>Electronic banking processes involve the use of digital data transmission techniques via electronic means and their processing by the financial institution. In some cases, such as the issuance and collection of receipts via the Internet, standards have been established that define and standardize the slips to be sent by the user to said institution.<br><br>Garcia at 2:1–14<br>However, no advances have been made in cases in which remittances are, for example, bank checks, |

| Claims | Prior Art |
| --- | --- |
| | because in these cases, physical documents must be present or sent to the respective bank branch for processing, whether manually or via reader terminals that capture certain data from the actual documents. We are not aware of a teleprocessing system that avoids the presence and custody of the document by the receiving institution in such a manner that allows the document validation process to be substantially accelerated, as well as its storage by electronic means, with the resulting advantages of speed in the process and storage capacity of the documents.<br><br>Garcia at 3:15–20<br>Document WO9736254 describes an automatic check processing system, which includes a device to input data related to the checking account and the amount, as well as means to capture a digitized image of the obverse of the check, which is sent electronically through a computer.<br><br>Garcia at 4:1–8<br>In summary, methodologies for electronic processing of documents and other banking information have been described, but none of these use the digital image of the document for storage in computer systems nor, in conjunction with the foregoing, means of emitting data by the user and the transmission thereof via a mobile telephone or similar portable equipment.<br><br>Garcia at 4:11–18<br>The present invention applies to the acceptance of documents, for example, banking or financial documents and preferably to the validation and deposit into the existing account at a banking institution of checks by a user using a mobile device that incorporates a camera, capable of capturing and transmitting digital images, as well as performing the usual telephony operations.<br><br>Garcia at 5:1–6:10<br>The method that is the subject-matter of the present invention is implemented in an interactive session conducted in real time, during which the user uses his mobile phone as a means of communicating and transmitting a document to the institution receiving said document, which it stores, processes, and, if applicable, validates automatically, informing the user of its acceptance or rejection of the transaction performed. In this process the following methodology is followed:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator. |

| Claims | Prior Art |
| --- | --- |
| | b) Entering various user recognition data, consisting of security keys for the user's identification and authentication, which may be a secret number, an associated telephone number and code (IMEI), as well as data associated with the document itself, such as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the previous points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 6:12–20<br>According to an important feature of the present invention, both the capture of images and the entry of data, the encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well as the usual telephony operations.<br><br>Garcia at 7:1–14<br>The data used for recognizing the user consist of those specific to the mobile communications operator, which may be the mobile telephone number, the mobile device number, and optionally, a global positioning system and those that are specific to the receiving institution, which may be authentication codes consisting of a username and password, or a secret number in any format. The combination of all these logical and physical elements implements an innovative security mechanism by combining identification and authentication elements specific to the receiving institution and to the mobile communication operator.<br><br>Garcia at 7:15–17<br>There is constant communication between the user and the financial institution via the mobile communications operator during the entire process.<br><br>Garcia at 7:18–8:14<br>This system achieves greater simplicity, speed, and security in the process of accepting bank documents and bills, and particularly in depositing checks, by replacing the traditional check reader with a multi- |

| Claims | Prior Art |
| --- | --- |
| | purpose mobile device. The immediate processing of the information transmitted by this system and the electronic treatment thereof allows the immediate availability in the user's account and the elimination of manual mechanization by the institution, which thus reduces the labor necessary because the truncation process is performed by remote devices external to the financial institution. The user of this service avoids unnecessary trips to the bank branch to present documents for collection, and the immediate display of the image in his account. In general, this system will facilitate the implementation of electronic banking in order to perform the transactions for which it is intended.<br><br>Garcia at 9:1–4<br>Figure 1 schematically depicts the implementation of the method of authentication and transmission of documents, the subject-matter of this invention.<br><br>Garcia at 9:6–7<br>Figure 2 depicts the usual means used in this authentication method.<br><br>Garcia at 9:9–18<br>Preferred embodiment of the Invention.<br>As can be observed in Figure 1, the method of accepting banking documents and bills with guarantee of origin, i.e., identifying and authenticating the issuer of the documents requires the use of a mobile device by the user of the service, a mobile communications services provider and also computer equipment at the institution that provides the service to the user.<br>Garcia at 9:19–10:18<br>The equipment required by the user must allow:<br>a) Identifying of the user, for which said user opens a computer application in the mobile telephone in which an interactive session is established with the receiving institution through the mobile communications operator.<br>b) Entering by the user of different recognition data consisting of username and password and selection of the beneficiary company.<br>c) Identifying the device in relation to the user, confirmation of the associated telephone number and the code (IMEI).<br>d) Capturing the digital image of the obverse and the reverse side of the document.<br>e) Capturing the data associated with the document, such as the amount and the check number. |

| Claims | Prior Art |
| --- | --- |
| | f) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>g) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br><br>Garcia at 11:1–6<br>The mobile equipment that performs the initial processes of capturing, digitizing, and processing the image consists of a mobile telephone equipped with the features and functions necessary for said purpose, particularly a mobile telephone with camera and processing capacity.<br><br>Garcia at 12:4–13:5<br>In a preferential embodiment, this method applies to authenticating the beneficiary of the bank check, the original document of which is digitized using a mobile device that incorporates a digital camera able to take a digital photograph of one or both sides of said document. Via the keypad the user enters the aforementioned identification data, as well as the amount of the check and all the information and codes necessary for the operability of the system, proceeding with sending it. Once the message is encrypted and accompanied by a digital certification through the mobile network (GSM, GPRS, or any future technology) of a telecommunications operator company. The institution that receives the message proceeds to validate and encrypt it, ultimately obtaining a digital photograph of the check and a data set that allows it to automatically process the deposit of the check in question into the service user's account. Finally, the method provides for the transmission of an SMS message or any other method to the user informing him that the transaction has been completed successfully or not and, if applicable, the status of the transaction performed, i.e., if it was deposited or rejected.<br><br>Garcia at 14:2–15:10<br>1- A method for the transmission, authentication and automatic processing of digitized documents, for example, financial and non-financial documents, characterized in that it comprises the following phases or steps:<br>a) Identifying the user, which opens an interactive session with the receiving institution through the mobile communications operator.<br>b) Entering various user recognition data, consisting of security keys for his identification and authentication, such as for example a secret number; recognition data of the mobile device, which may be an associated telephone number and code (IMEI); as well as data associated with the document itself, such |

| Claims | Prior Art |
|---|---|
| | as the amount and/or the document number.<br>c) Capturing at least one digital image of the document.<br>d) Digitally encrypting and certifying the data and/or images captured in the preceding points.<br>e) Transmitting the digital image and the recognition data, once encrypted, to the institution that must accept the document.<br>f) The immediate displaying of the digital image by the user.<br>g) Recognizing, verifying, and electronically treating and processing the information received by said institution.<br><br>Garcia at 15:12–20<br>2.- The method according to claim 1, characterized in that both the capture of images and the entry of data, encryption thereof and the transmission of data and images and the other operations described in sections a), b), c), d), and e) of claim 1 are performed with a mobile device with incorporated digital camera, able to capture and transmit digital images, as well the usual telephony operations.<br><br>Garcia at 16:17–22<br>4.- The method according to any of the preceding claims, characterized in that it is applied preferably to the validation and deposit in a checking account in a banking institution by a user directly from his mobile telephone via the mobile communications operator.<br><br>*See* limitations 1[a], 1[h], 1[j].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14b] providing instructions to the user for capturing the check; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing the check." For example:<br><br>*See* limitation 1[b].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [14c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14d] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14e] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14f] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14g] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14h] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 10[1h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14i] performing image processing operations on the image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "performing image processing operations on the image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device." For example:<br>*See* limitations 1[h], 1[k].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device; and | |
| [14j] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit, | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14k] wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising: | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations." For example:<br>*See* limitation 1[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14l] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 1[n]. |

| Claims | Prior Art |
|---|---|
| satisfy the at least one monitoring criterion; and | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14m] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 1[p].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 15[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15a] receiving an image of a check, wherein the image of the check is provided for mobile check deposit in a bank by a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer and a downloaded app, the downloaded app associated with the bank to control check deposit by controlling the camera through the software abstraction layer, and the | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving an image of a check, wherein the image of the check is provided for mobile check deposit in a bank by a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer and a downloaded app, the downloaded app associated with the bank to control check deposit by controlling the camera through the software abstraction layer, and the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 14[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 716 of 1772

| Claims | Prior Art |
|---|---|
| downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | |
| [15b] providing instructions to the user for capturing the check; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15d] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15e] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example: |

| Claims | Prior Art |
|---|---|
| automatically capturing the video frame; | *See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15f] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15g] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15h] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 10[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [15i] performing image processing operations on the image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device; and | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "performing image processing operations on the image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device." For example:<br>*See* limitation 14[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15j] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit, | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 14[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15k] wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* limitation 2[m]. |

| Claims | Prior Art |
|---|---|
| spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 16. The method of claim 15, wherein the plurality of monitoring criteria includes a criterion based on magnetic ink character recognition (MICR) line detection. | Garcia discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 15, wherein the plurality of monitoring criteria includes a criterion based on magnetic ink character recognition (MICR) line detection." For example:<br>*See* claim 3.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Garcia alone; (2) Garcia in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

**Exhibit A18 - Invalidity of 12,159,310 in view of Ranjan**

Using a Camera with Windows Mobile 5 by Amit Ranjan ("Ranjan") anticipates and/or renders obvious (alone or in combination with other references), at least under USAA's apparent infringement theories, all asserted claims of U.S. Patent No. 12,159,310 ("the '310 patent"). Ranjan was published or otherwise publicly available by July 21, 2006 and is therefore prior art to the '310 patent.

Regions has charted the portions of the reference that correspond to each element, including at least under USAA's apparent interpretation of the claims as reflected in Regions Complaint and Infringement Contentions. Where the chart below states that the reference "discloses," "teaches," or "suggests" a claim limitation, such disclosure, teaching, or suggestion may be express, implied, or inherent. The cited evidence is merely illustrative, and Regions reserves the right to cite alternative or additional evidence.

No part of this exemplary chart construes, or is intended to construe, the specification, file history, or claims of the asserted patents. In addition, nothing stated in this chart shall be treated as an admission or suggestion that Regions agrees with USAA regarding either the scope of any of the asserted claims or Regions' apparent interpretation of the scope of the asserted claims, or that Regions' accused products meet any limitations of the claims.

To the extent that USAA contends that Ranjan does not disclose one or more limitations of the asserted claims, it would have been obvious to combine the teachings of Ranjan with (1) the knowledge of one of ordinary skill in the art, and/or (2) the teachings of any of the prior art set forth in Regions' other invalidity charts and cover pleading, to show all the limitations of the asserted claims, including for the reasons set forth in Regions' Invalidity Contention Cover Pleading, which are incorporated herein.

The cited disclosure further shows that it would have been obvious to combine Ranjan with the prior art set forth in Regions' other invalidity charts and cover pleading, for example by showing similarities with the other prior art and/or elements that could be improved by combination, replacement, and/or modification in view of the other prior art.

Because USAA has yet to identify any limitation of the asserted claims that it contends is not fully disclosed by Ranjan, either alone or in combination with other prior art cited by Regions, Regions expressly reserves the right to rebut any such contention, including by identifying additional § 103 combinations, if any such contention is made by USAA.

| **Claims** | **Prior Art** |
|---|---|
| 1[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example: |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 721 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br><br>Ranjan at Page 1<br>  <br> |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3<br>Building the Graph<br><br>A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.<br><br>First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:<br><br>`HRESULT hResult = S_OK;`<br><br>`IGraphBuilder *pFilterGraph;` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 724 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

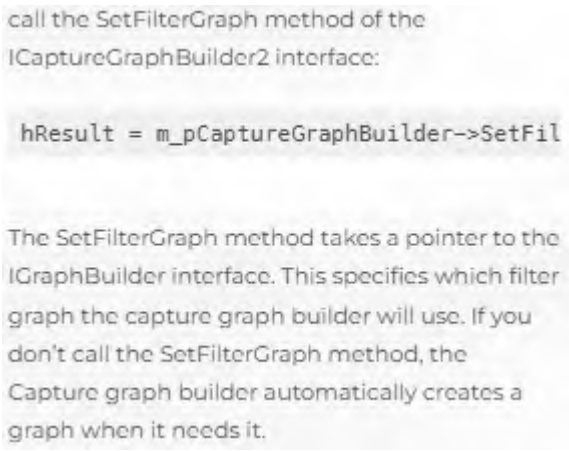
                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 725 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 729 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |

The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 732 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 733 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



***Note:*** *WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

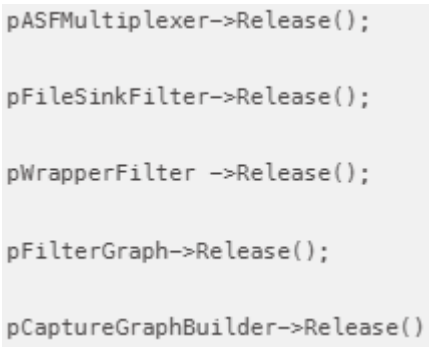
//Stop capturing

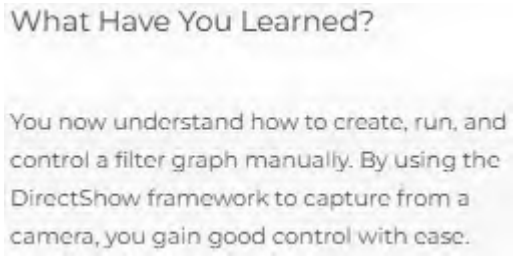
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 734 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13   |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1a] providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer for controlling the camera, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations comprising: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer for controlling the camera, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br><br>Ranjan at Page 1<br>  <br> |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 740 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

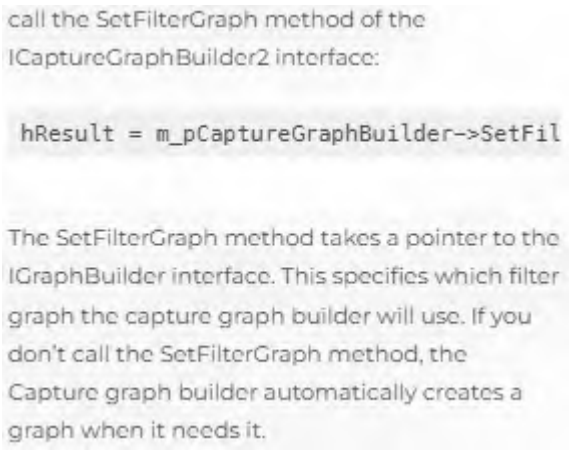
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

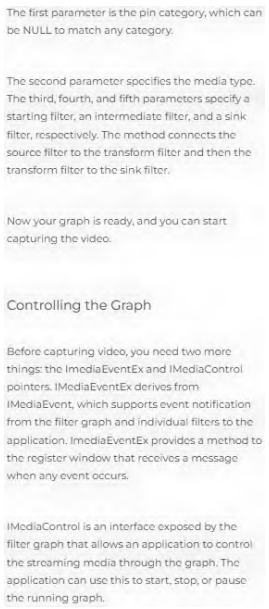
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 749 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

//Stop capturing

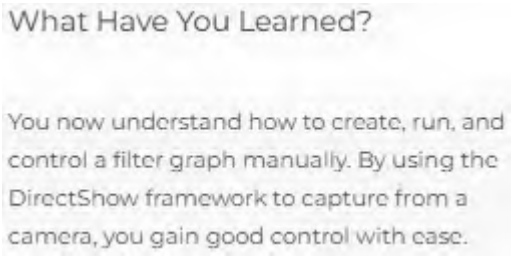
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 751 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1b] providing instructions to the user for capturing an image of a check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of a check." For example:<br>Ranjan at page 1<br>  <br> |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at Page 1<br>  <br><br>Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 755 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

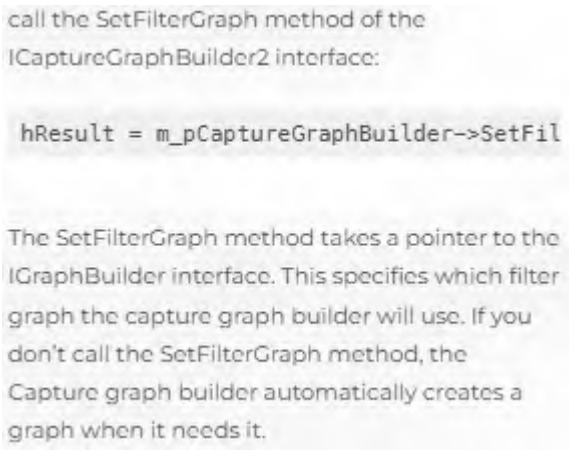
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8 |

Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:

```
IBaseFilter *pASFMultiplexer;

IFileSinkFilter *pFileSinkFilter;

hResult = pCaptureGraphBuilder->SetOutpu
```

https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

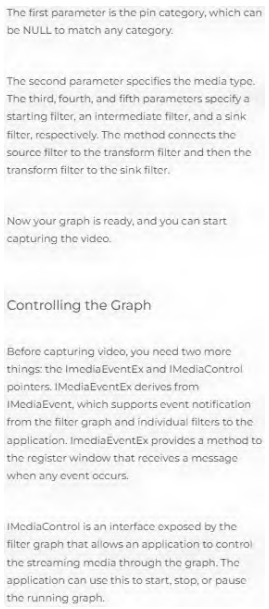
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 764 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

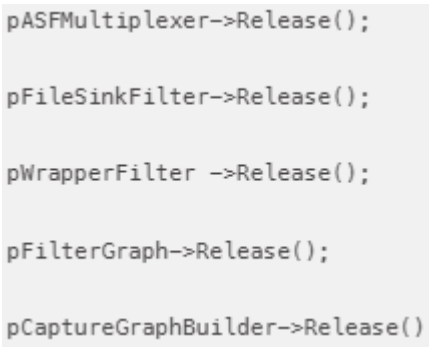
//Stop capturing

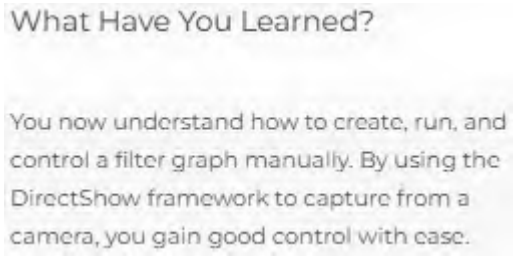
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 765 of 1772

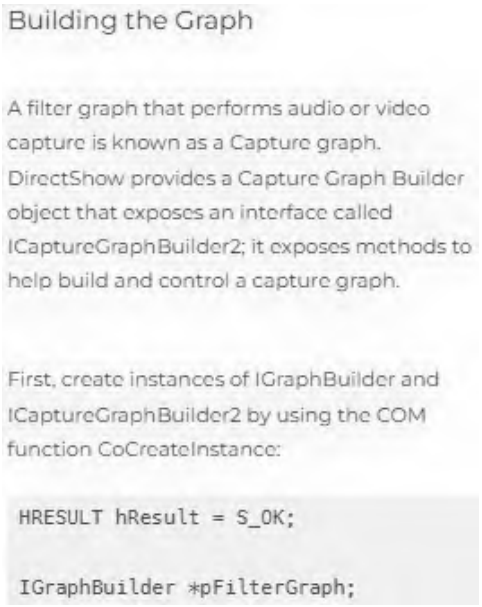
| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13   |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>PVideoCaptureFilter->Release ();<br><br>pVideoEncoder->Release ();<br><br>pMediaEvent ->Release();<br><br>pMediaSeeking ->Release();<br><br>Ranjan at page 17<br>  <br>pASFMultiplexer->Release();<br><br>pFileSinkFilter->Release();<br><br>pWrapperFilter ->Release();<br><br>pFilterGraph->Release();<br><br>pCaptureGraphBuilder->Release() |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Figure 1: Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3<br>  <br>Building the Graph<br><br>A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.<br><br>First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:<br><br>`HRESULT hResult = S_OK;`<br><br>`IGraphBuilder *pFilterGraph;` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 775 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8 |

Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:

```
IBaseFilter *pASFMultiplexer;

IFileSinkFilter *pFileSinkFilter;

hResult = pCaptureGraphBuilder->SetOutpu
```

https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

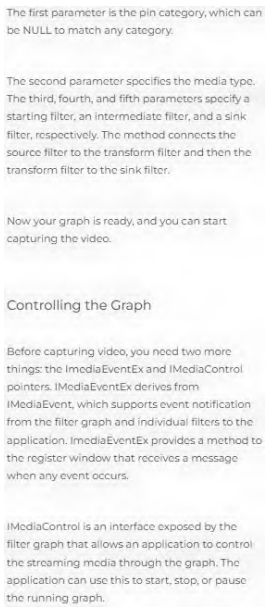
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 780 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

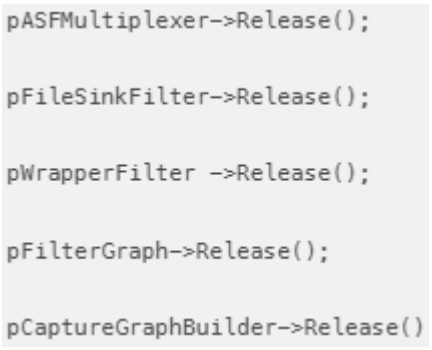
//Stop capturing

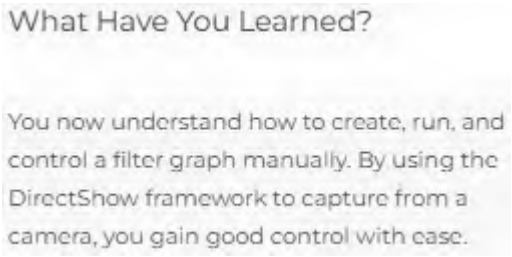
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |

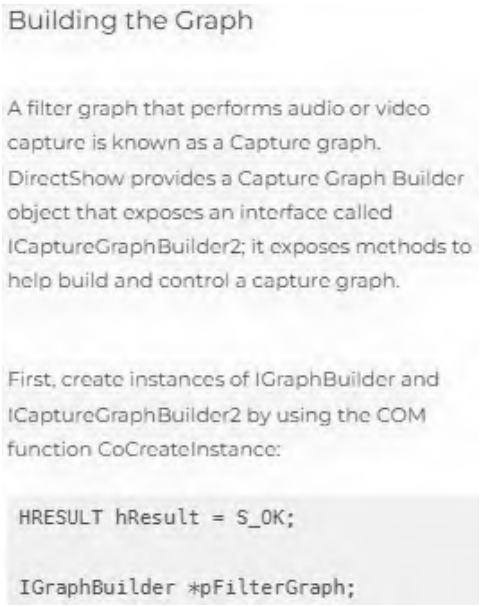

USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 782 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>PVideoCaptureFilter->Release ();<br><br>pVideoEncoder->Release ();<br><br>pMediaEvent ->Release();<br><br>pMediaSeeking ->Release();<br><br>Ranjan at page 17<br>  <br>pASFMultiplexer->Release();<br><br>pFileSinkFilter->Release();<br><br>pWrapperFilter ->Release();<br><br>pFilterGraph->Release();<br><br>pCaptureGraphBuilder->Release() |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1d] overlaying an alignment guide on the live video of the check in the field of view of the camera; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>Ranjan at page 1<br>  <br><br>**Figure 1:** Filter Graph for Video Capture |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at Page 1<br>  <br><br>Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |



Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 786 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

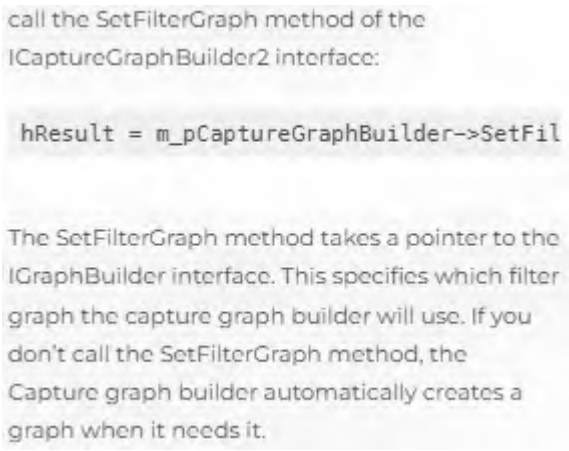
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 790 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

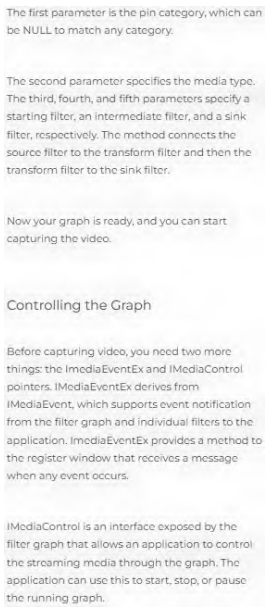
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

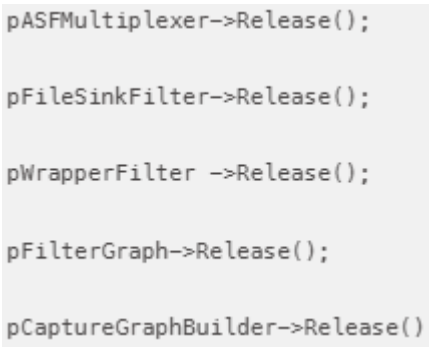
//Stop capturing

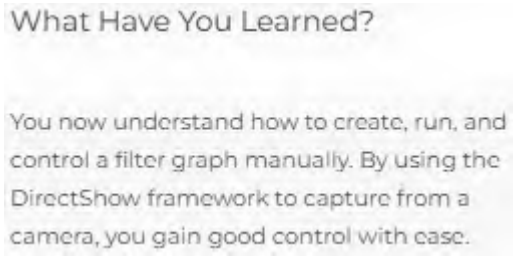
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 796 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13   |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>PVideoCaptureFilter->Release ();<br>pVideoEncoder->Release ();<br>pMediaEvent ->Release();<br>pMediaSeeking ->Release();<br><br>Ranjan at page 17<br>  <br>pASFMultiplexer->Release();<br>pFileSinkFilter->Release();<br>pWrapperFilter ->Release();<br>pFilterGraph->Release();<br>pCaptureGraphBuilder->Release() |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitation 1[pre], 1[a], 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1e] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br><br>Ranjan at Page 1<br>  <br> |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 802 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

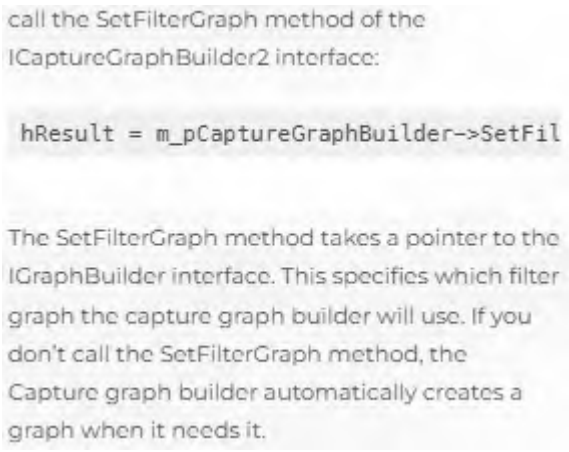
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

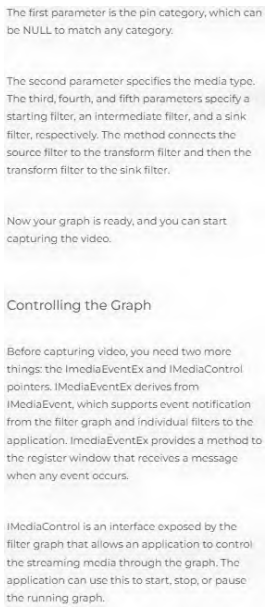
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

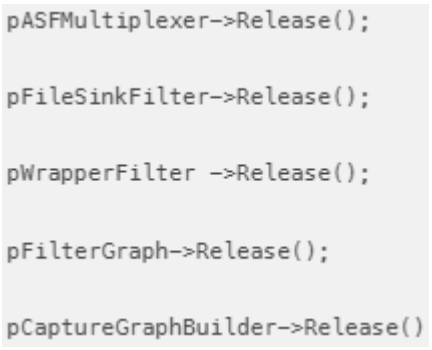
//Stop capturing

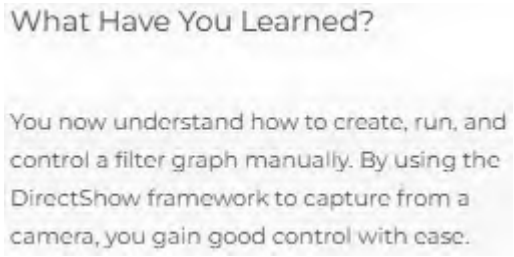
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 812 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>PVideoCaptureFilter->Release ();<br>pVideoEncoder->Release ();<br>pMediaEvent ->Release();<br>pMediaSeeking ->Release();<br><br>Ranjan at page 17<br>  <br>pASFMultiplexer->Release();<br>pFileSinkFilter->Release();<br>pWrapperFilter ->Release();<br>pFilterGraph->Release();<br>pCaptureGraphBuilder->Release() |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[a], 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br><br>Ranjan at Page 1<br>  <br> |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 818 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8 |

Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:

```
IBaseFilter *pASFMultiplexer;

IFileSinkFilter *pFileSinkFilter;

hResult = pCaptureGraphBuilder->SetOutpu
```

https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

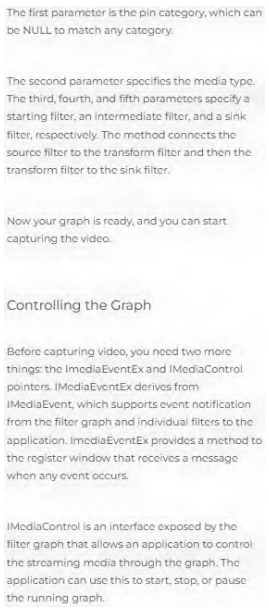
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

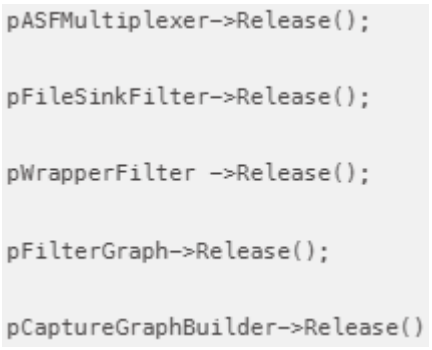
   wStartCookie, wEndCookie );

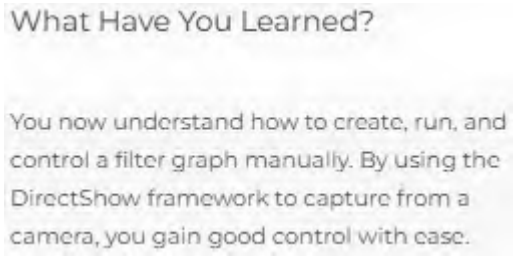
//Stop capturing

dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13   |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>PVideoCaptureFilter->Release ();<br><br>pVideoEncoder->Release ();<br><br>pMediaEvent ->Release();<br><br>pMediaSeeking ->Release();<br><br>Ranjan at page 17<br>  <br>pASFMultiplexer->Release();<br><br>pFileSinkFilter->Release();<br><br>pWrapperFilter ->Release();<br><br>pFilterGraph->Release();<br><br>pCaptureGraphBuilder->Release() |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[a], 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Figure 1: Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3<br>Building the Graph<br><br>A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.<br><br>First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:<br><br>`HRESULT hResult = S_OK;`<br><br>`IGraphBuilder *pFilterGraph;` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 834 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 838 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

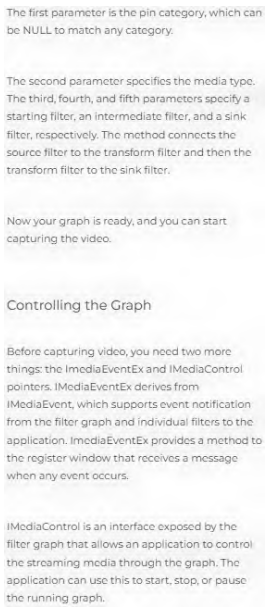
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

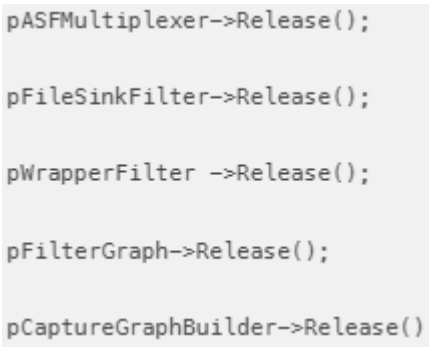
   wStartCookie, wEndCookie );

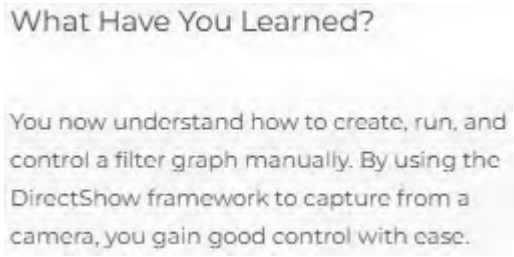
//Stop capturing

dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>PVideoCaptureFilter->Release ();<br>pVideoEncoder->Release ();<br>pMediaEvent ->Release();<br>pMediaSeeking ->Release();<br><br>Ranjan at page 17<br>  <br>pASFMultiplexer->Release();<br>pFileSinkFilter->Release();<br>pWrapperFilter ->Release();<br>pFilterGraph->Release();<br>pCaptureGraphBuilder->Release() |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[a], 1[e], 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1h] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Figure 1: Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 850 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 854 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

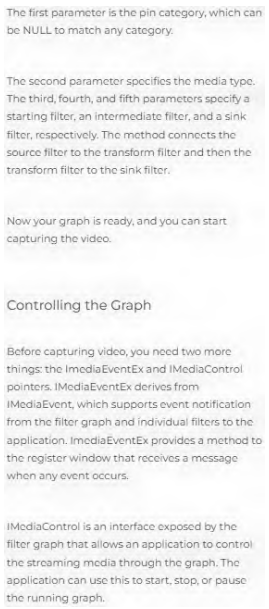
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

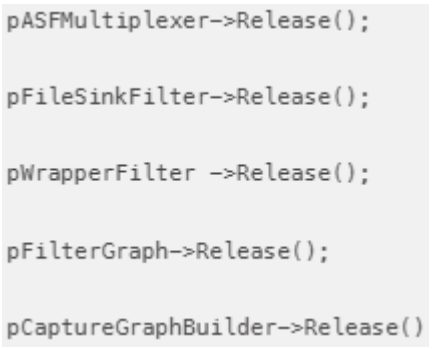
   wStartCookie, wEndCookie );

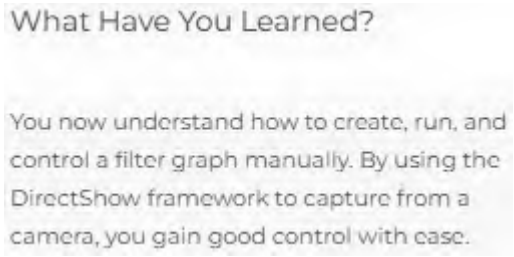
//Stop capturing

dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>PVideoCaptureFilter->Release ();<br><br>pVideoEncoder->Release ();<br><br>pMediaEvent ->Release();<br><br>pMediaSeeking ->Release();<br><br>Ranjan at page 17<br>  <br>pASFMultiplexer->Release();<br><br>pFileSinkFilter->Release();<br><br>pWrapperFilter ->Release();<br><br>pFilterGraph->Release();<br><br>pCaptureGraphBuilder->Release() |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[a], 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1i] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Capture<br>Video Encoder<br>Multi – Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 866 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

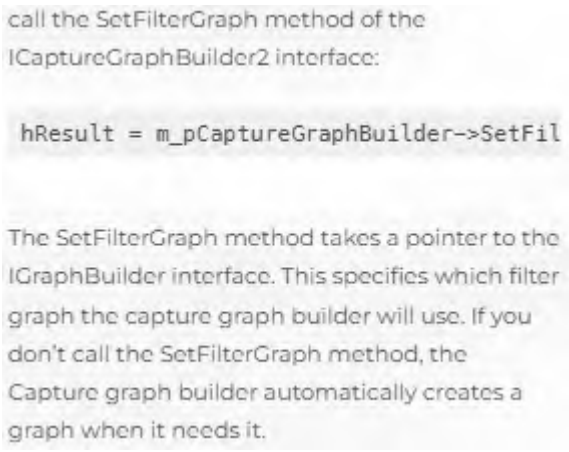
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 870 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8 |

Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:

```
IBaseFilter *pASFMultiplexer;

IFileSinkFilter *pFileSinkFilter;

hResult = pCaptureGraphBuilder->SetOutpu
```

https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

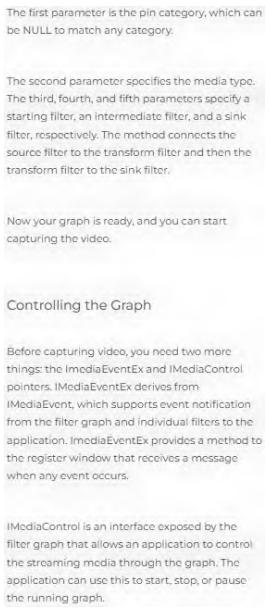
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

//Stop capturing

dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

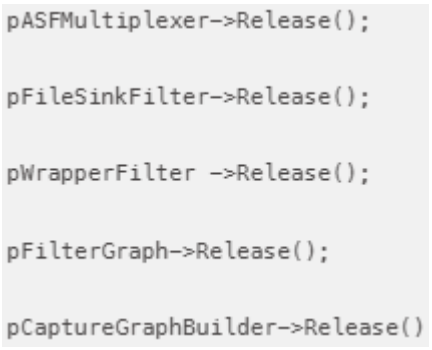
   &dwStart, &dwEnd, wStartCookie, wEndC
```

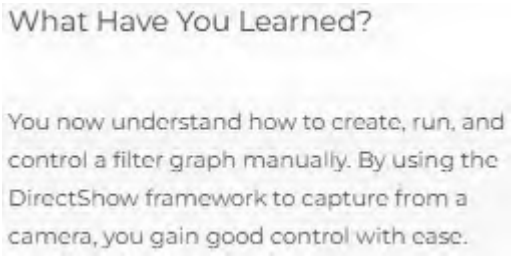
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.

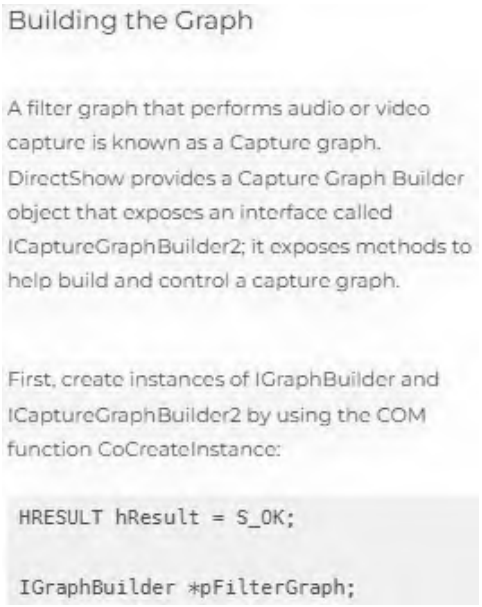

USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 877 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1j] receiving the digital image of the check transmitted by the user's mobile device; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving the digital image of the check transmitted by the user's mobile device." For example:<br>Ranjan at page 1<br>  <br><br>**Figure 1:** Filter Graph for Video Capture |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at Page 1<br>  <br><br>Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |



Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 881 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 885 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8 |

Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:

```
IBaseFilter *pASFMultiplexer;

IFileSinkFilter *pFileSinkFilter;

hResult = pCaptureGraphBuilder->SetOutpu
```

https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

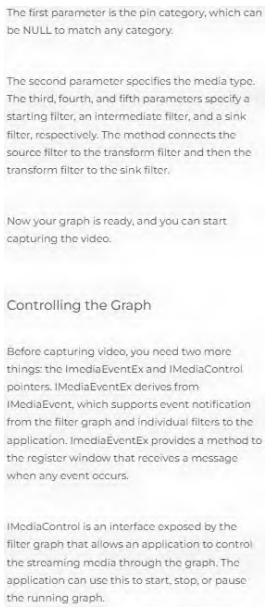
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

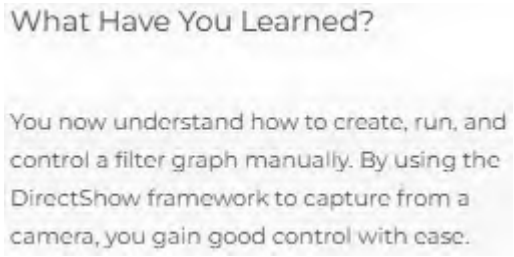
//Stop capturing

dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[h], 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1k] processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Figure 1: Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3<br>Building the Graph<br><br>A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.<br><br>First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:<br><br>`HRESULT hResult = S_OK;`<br><br>`IGraphBuilder *pFilterGraph;` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 901 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

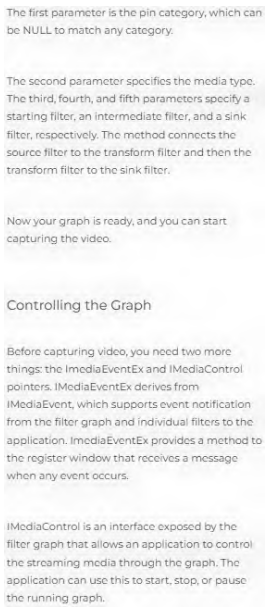
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

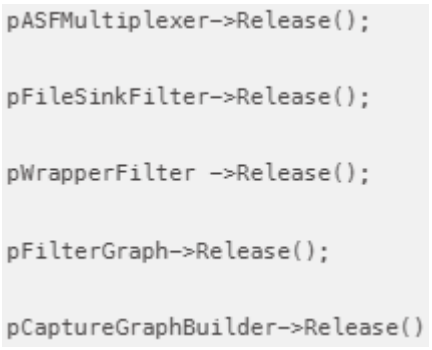
   wStartCookie, wEndCookie );

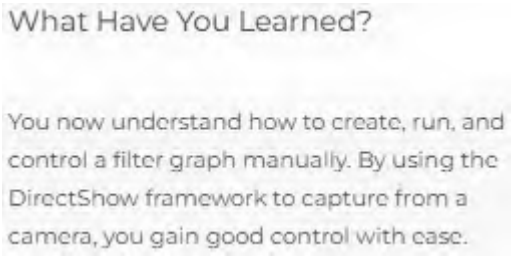
//Stop capturing

dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |

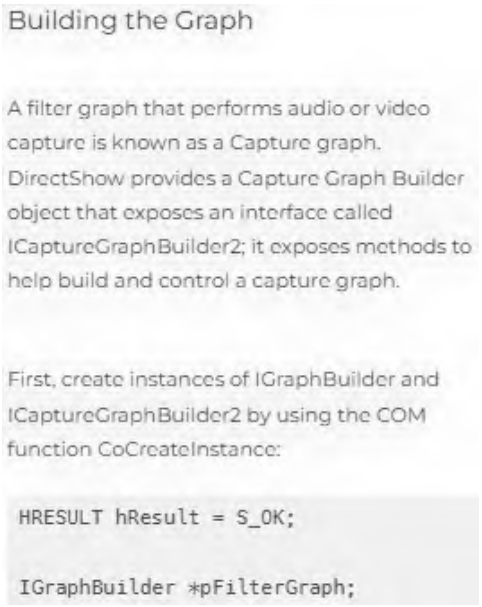
| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1l] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit, | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |



Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

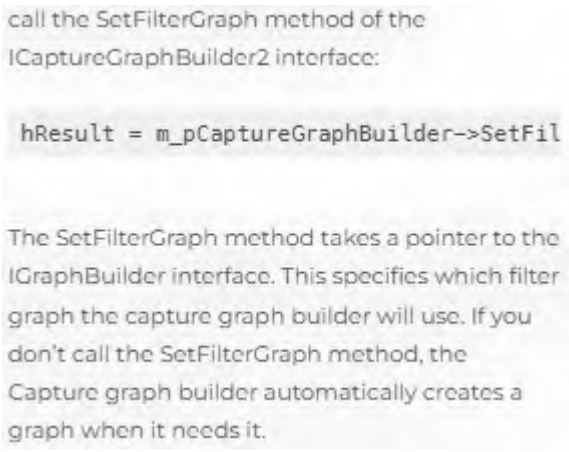
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 917 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8 |

Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:

```
IBaseFilter *pASFMultiplexer;

IFileSinkFilter *pFileSinkFilter;

hResult = pCaptureGraphBuilder->SetOutpu
```

https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

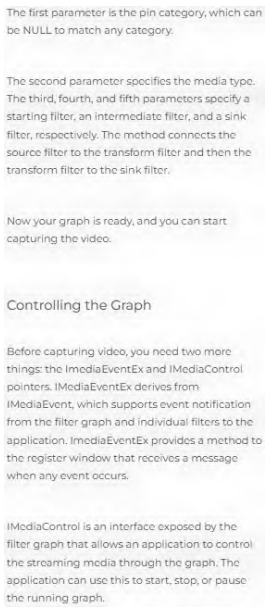
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

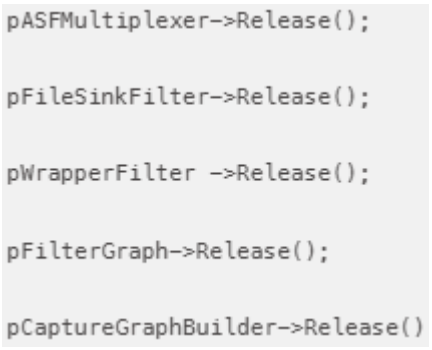
   wStartCookie, wEndCookie );

//Stop capturing

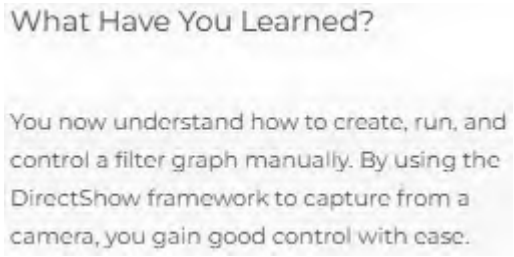
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>PVideoCaptureFilter->Release ();<br><br>pVideoEncoder->Release ();<br><br>pMediaEvent ->Release();<br><br>pMediaSeeking ->Release();<br><br>Ranjan at page 17<br>  <br>pASFMultiplexer->Release();<br><br>pFileSinkFilter->Release();<br><br>pWrapperFilter ->Release();<br><br>pFilterGraph->Release();<br><br>pCaptureGraphBuilder->Release() |

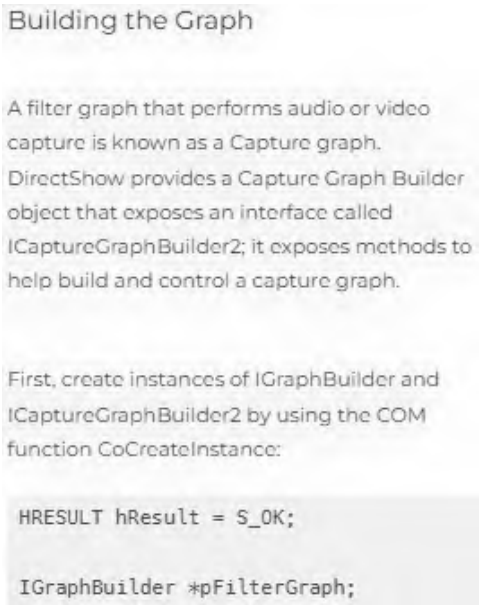

USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 925 of 1772

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[h], 1[i], 1[k].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1m] wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Figure 1: Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |



Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 929 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

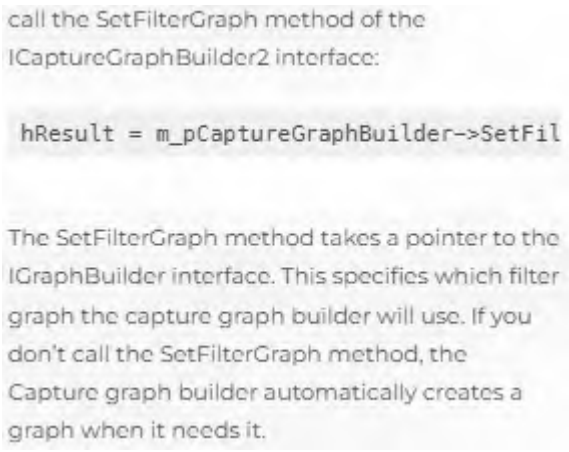
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 933 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8 |

Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:

```
IBaseFilter *pASFMultiplexer;

IFileSinkFilter *pFileSinkFilter;

hResult = pCaptureGraphBuilder->SetOutpu
```

https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |

The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 937 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 938 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

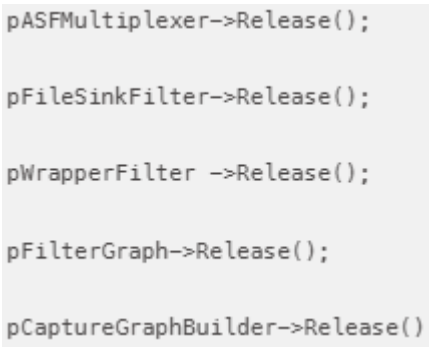
//Stop capturing

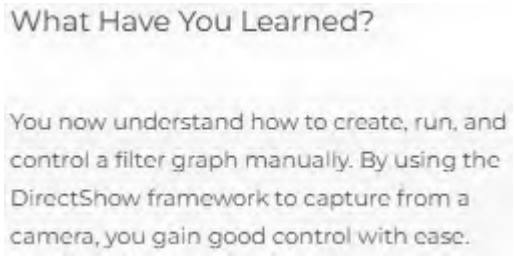
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13   |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 940 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>PVideoCaptureFilter->Release ();<br>pVideoEncoder->Release ();<br>pMediaEvent ->Release();<br>pMediaSeeking ->Release();<br><br>Ranjan at page 17<br>  <br>pASFMultiplexer->Release();<br>pFileSinkFilter->Release();<br>pWrapperFilter ->Release();<br>pFilterGraph->Release();<br>pCaptureGraphBuilder->Release() |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1n] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Capture<br>Video Encoder<br>Multi – Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 945 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 949 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

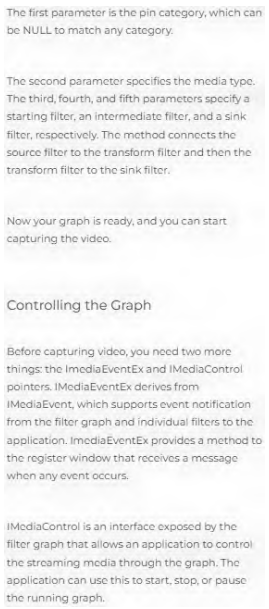
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 954 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

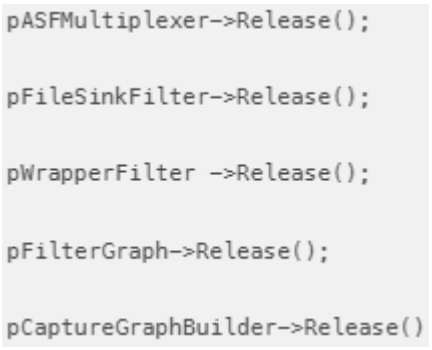
//Stop capturing

dwStart=0;

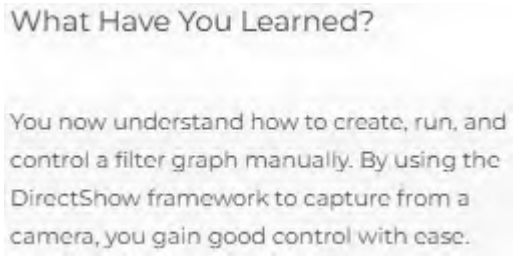
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 956 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>PVideoCaptureFilter->Release ();<br><br>pVideoEncoder->Release ();<br><br>pMediaEvent ->Release();<br><br>pMediaSeeking ->Release();<br><br>Ranjan at page 17<br>  <br>pASFMultiplexer->Release();<br><br>pFileSinkFilter->Release();<br><br>pWrapperFilter ->Release();<br><br>pFilterGraph->Release();<br><br>pCaptureGraphBuilder->Release() |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 957 of 1772

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[a],1[e], 1[f], 1[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1o] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Figure 1: Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 961 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 965 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |

The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 969 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 970 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

//Stop capturing

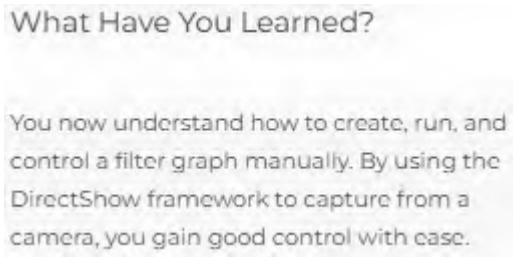
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 971 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[a], 1[b], 1[e], 1[f], 1[h], 1[m], 1[n].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 2[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2a] providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer for controlling the camera, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer for controlling the camera, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| user's mobile device, is configured to cause the user's mobile device to perform operations comprising: | |
| [2b] providing instructions to the user for capturing an image of a check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[b]<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2d] overlaying an alignment guide on the live video of the check in the field of view of the camera; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2e] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

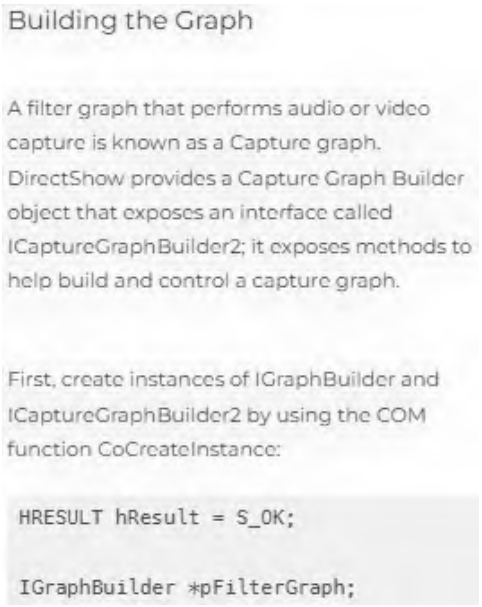
| Claims | Prior Art |
|---|---|
| [2f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2h] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2i] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 1[i]. |

| Claims | Prior Art |
|---|---|
| character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2j] receiving the digital image of the check transmitted by the user's mobile device; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving the digital image of the check transmitted by the user's mobile device." For example:<br>*See* limitation 1[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2k] processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations including receiving an indication of a result of the processing before confirming that the mobile check deposit can go forward." For example:<br>*See* limitation 1[k].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2l] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit." For example: |

| Claims | Prior Art |
|---|---|
| check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit, | *See* limitation 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2m] wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at Page 1<br>  <br><br>Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |



Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 980 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

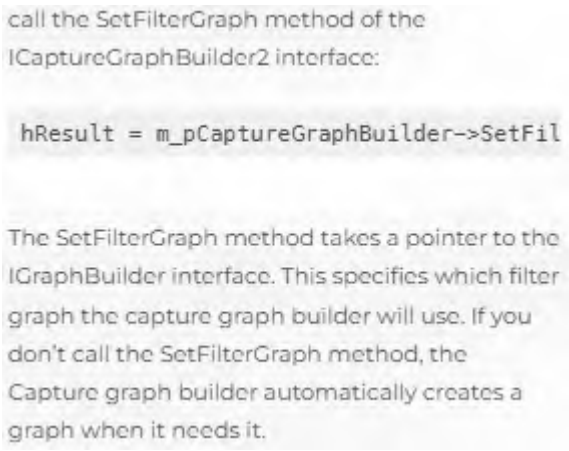
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 984 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

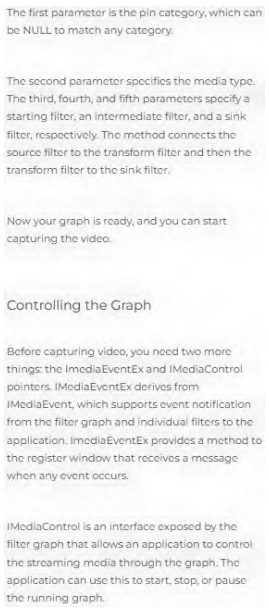
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 989 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

//Stop capturing

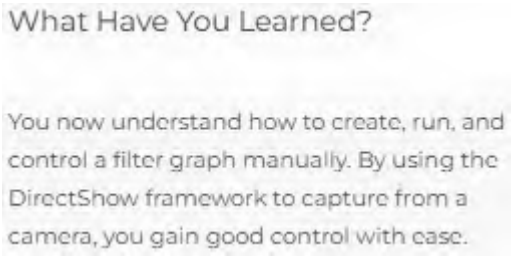
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 990 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13   |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 3. The method of claim 2, wherein the plurality of monitoring criteria includes a criterion based on magnetic ink character recognition (MICR) line detection. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 2, wherein the plurality of monitoring criteria includes a criterion based on magnetic ink character recognition (MICR) line detection." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Figure 1: Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 996 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

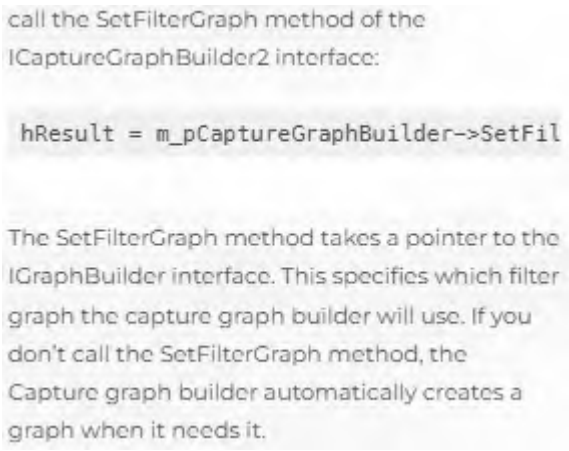
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1000 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

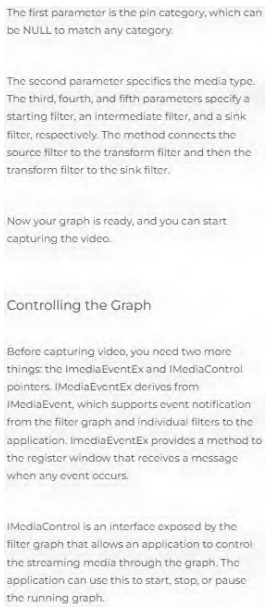
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1005 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

//Stop capturing

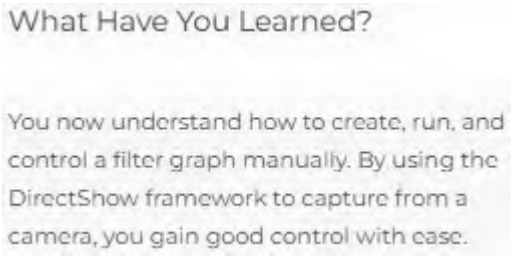
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1007 of 1772

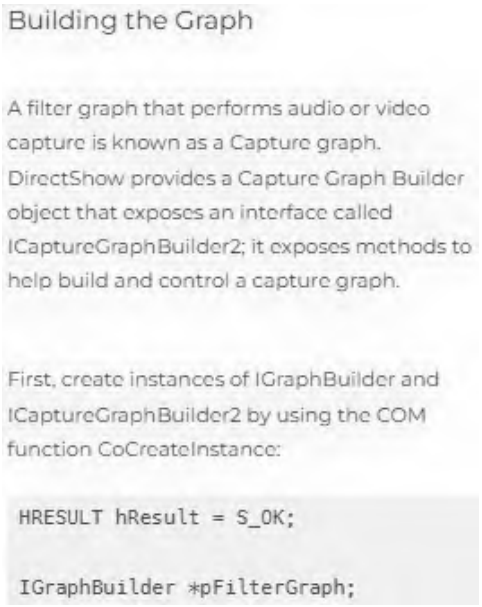
| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitation 2[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 4. The method of claim 2, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 2, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Figure 1: Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |



Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1012 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

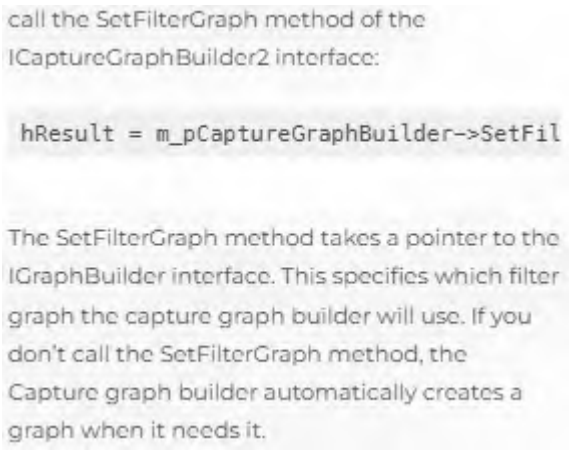
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

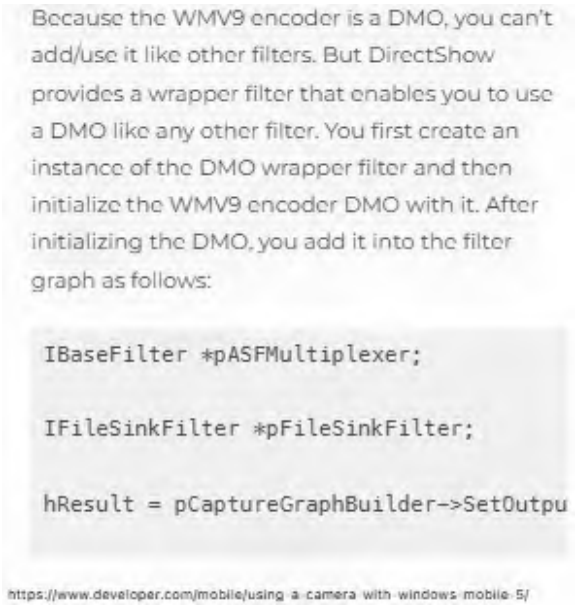
| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1016 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

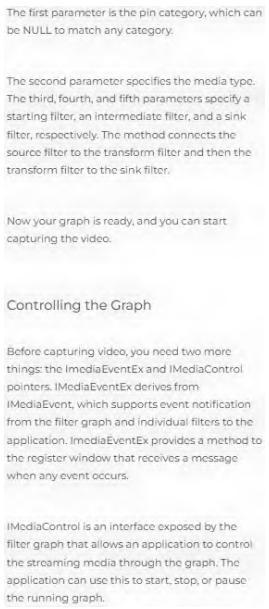
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1020 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

//Stop capturing

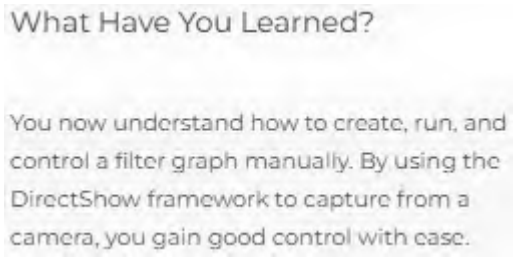
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1023 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitation 2[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 5[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5a] monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitations 1[a], 1[c], 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | |
| [5b] providing instructions to the user for capturing an image of the check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5c] generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5d] overlaying an alignment guide on the live video of the check in the field of view of the camera; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5e] using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured." For example:<br>*See* limitations 1[e]–1[h]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5h] processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitations 1[k], 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5i] confirming that the mobile check deposit can go forward after optical character recognition is | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in |

| Claims | Prior Art |
|---|---|
| performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; | the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5j] updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5k] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 1[n].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [5l] after determining that the different video frame of the live video of the check does not satisfy | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example: |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1028 of 1772

| Claims | Prior Art |
|---|---|
| the at least one monitoring criterion, providing feedback to the user. | *See* limitation 1[o].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 6[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6a] monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 5[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6b] providing instructions to the user for capturing an image of the check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of the check." For example:<br>*See* limitation 1[b]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6c] generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6d] overlaying an alignment guide on the live video of the check in the field of view of the camera; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6e] using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6h] processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitation 5[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6i] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 5[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [6j] updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit, | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [6k] wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* limitation 2[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 7. The method of claim 6, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 6, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping." For example:<br>*See* claim 4.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| 8[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8a] monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 5[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8b] providing instructions to the user for capturing an image of the check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [8c] generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8d] overlaying an alignment guide on the live video of the check in the field of view of the camera; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8e] using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured." For example:<br>*See* limitation 5[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8g] when the video frame of the live video of the check in the field | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1034 of 1772

| Claims | Prior Art |
|---|---|
| of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8h] processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitation 5[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8i] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 5[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8j] updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 5[j]. |

| Claims | Prior Art |
|---|---|
| the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit, | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [8k] wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising the monitoring, the determining, the providing, and the confirming. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising the monitoring, the determining, the providing, and the confirming." For example:<br>*See* limitations 1[a], 1[b], 1[f]–1[h], 5[a], 5[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 9[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | *See* limitation 1[pre].<br>To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9a] monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring a live video of a check in a field of view of a camera with respect to at least one monitoring criterion, wherein the live video is generated by a mobile device of a user, the user's mobile device including the camera and a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 5[a]. |

| Claims | Prior Art |
|---|---|
| control check deposit by controlling the camera, the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9b] providing instructions to the user for capturing an image of the check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9c] generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating the live video of the check in the field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9d] overlaying an alignment guide on the live video of the check in the field of view of the camera; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9e] using a wireless network, submitting the check for mobile check deposit in the bank after a | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, submitting the check for mobile check deposit in the bank after a frame of the live video of the check in the field of view of the camera is automatically captured." For example: |

| Claims | Prior Art |
|---|---|
| frame of the live video of the check in the field of view of the camera is automatically captured; | *See* limitations 5[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9h] processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing a digital image of the check including performing optical character recognition and verifying that the digital image of the check is acceptable for mobile check deposit, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitation 5[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9i] confirming that the mobile check deposit can go forward after | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in |

| Claims | Prior Art |
|---|---|
| optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; and | the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 5[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9j] updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit, | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a bank computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 5[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9k] wherein the confirming step is performed before the submitting step and the user's mobile device receives an indication of a result of the processing the digital image of the check before the confirming step. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the confirming step is performed before the submitting step and the user's mobile device receives an indication of a result of the processing the digital image of the check before the confirming step." For example:<br>*See* limitation 5[i], 1[i], 1[k].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 10[pre]. A method enabling mobile check deposit auto-capture functionality via video frame | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example: |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1039 of 1772

| Claims | Prior Art |
|---|---|
| processing, the method comprising: | *See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10a] providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations comprising: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10b] providing instructions to the user for capturing an image of a check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10d] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10e] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and the automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and the automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10f] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10g] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example: |

| Claims | Prior Art |
|---|---|
| for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | *See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10h] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10i] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward, | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward." For example:<br>*See* limitation 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10j] wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the app, when downloaded and executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations." For example:<br>*See* limitation 1[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [10k] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 1[n].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10l] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 1[o].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 11[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11a] providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer, the app associated with a bank to control check deposit by controlling the camera through the | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing an app for download to a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer, the app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 10[a]. |

| Claims | Prior Art |
|---|---|
| software abstraction layer, wherein the app, when downloaded to the user's mobile device and executed by the user's mobile device, is configured to cause the user's mobile device to perform operations comprising: | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11b] providing instructions to the user for capturing an image of a check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11d] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [11e] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11f] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11g] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11h] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 10[h]. |

| Claims | Prior Art |
|---|---|
| (MICR) line on the check is recognized, and an amount payable on the check is validated; and | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11i] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward, | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward." For example:<br>*See* limitation 10[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [11j] wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* limitation 2[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 12. The method of claim 11, wherein the plurality of monitoring criteria being based on magnetic ink character recognition (MICR) line detection and one or both of corner detection and edge detection. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 11, wherein the plurality of monitoring criteria being based on magnetic ink character recognition (MICR) line detection and one or both of corner detection and edge detection." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1049 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

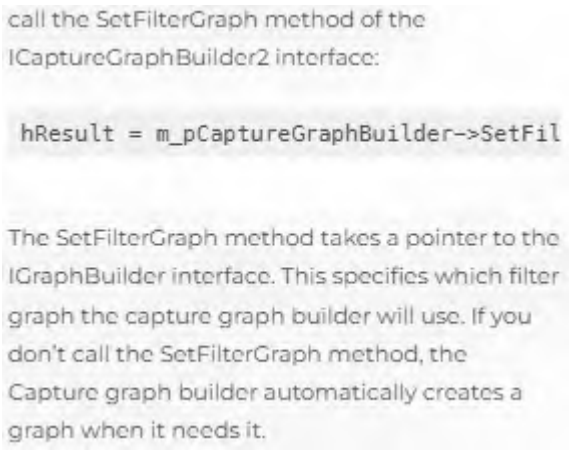
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1053 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

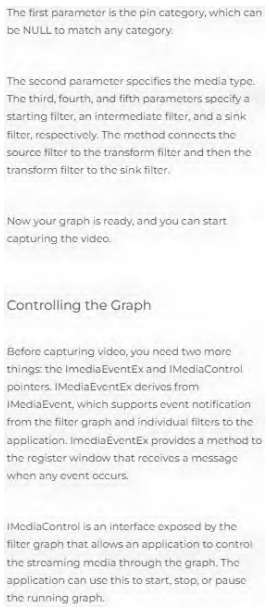
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1058 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

//Stop capturing

dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |



```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

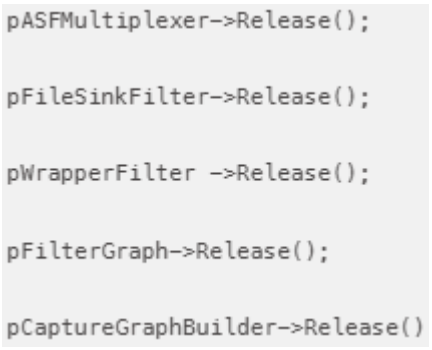
   &dwStart, &dwEnd, wStartCookie, wEndC
```

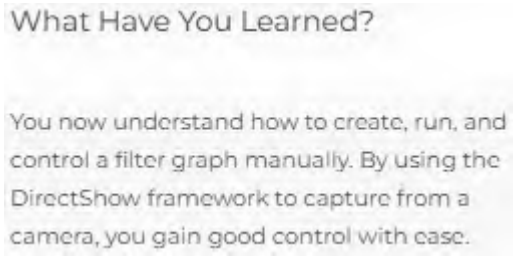
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

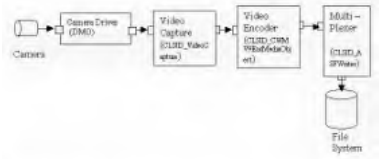
Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1060 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br><br>PVideoCaptureFilter->Release ();<br>pVideoEncoder->Release ();<br>pMediaEvent ->Release();<br>pMediaSeeking ->Release();<br><br>Ranjan at page 17<br>  <br>pASFMultiplexer->Release();<br>pFileSinkFilter->Release();<br>pWrapperFilter ->Release();<br>pFilterGraph->Release();<br>pCaptureGraphBuilder->Release() |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See* claims 3, 4.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 13. The method of claim 11, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 11, wherein the plurality of monitoring criteria includes criteria based on light brightness, skewing, and warping." For example:<br>*See* claim 4.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 14[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14a] receiving an image of a check, wherein the image of the check is provided for mobile check | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving an image of a check, wherein the image of the check is provided for mobile check deposit in a bank by a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction |

| Claims | Prior Art |
| --- | --- |
| deposit in a bank by a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer and a downloaded app, the downloaded app associated with the bank to control check deposit by controlling the camera through the software abstraction layer, and the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | layer and a downloaded app, the downloaded app associated with the bank to control check deposit by controlling the camera through the software abstraction layer, and the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br><br>Ranjan at page 1<br><br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br><br>  <br><br>**Figure 1:** Filter Graph for Video Capture<br><br>Ranjan at Page 1<br><br>  <br> |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1065 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

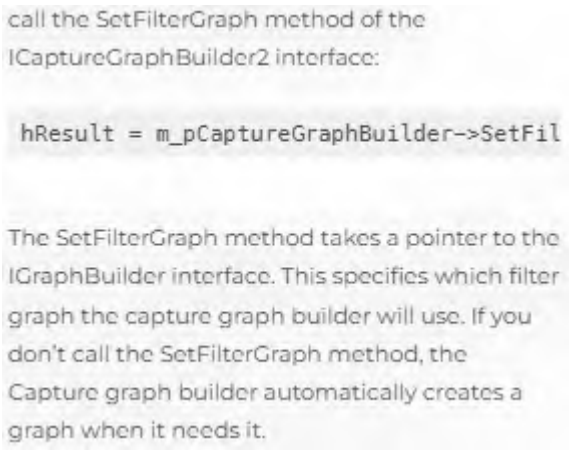
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1069 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8 |

Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:

```
IBaseFilter *pASFMultiplexer;

IFileSinkFilter *pFileSinkFilter;

hResult = pCaptureGraphBuilder->SetOutpu
```

https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

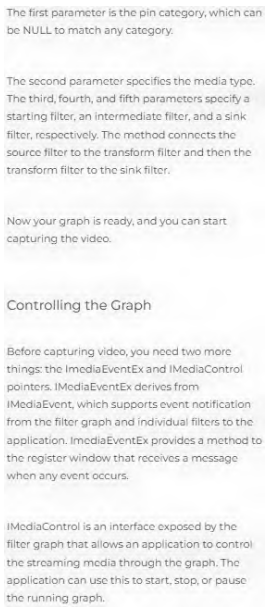
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10<br>  <br>The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0):

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                     WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1074 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;

WORD wStartCookie = 1, wEndCookie = 2;

dwEnd=MAXLONGLONG

//start capturing

hResult=pCaptureGraphBuilder->ControlStr

   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

   pVideoCaptureFilter, &dwStart, &dwEnd

   wStartCookie, wEndCookie );

//Stop capturing

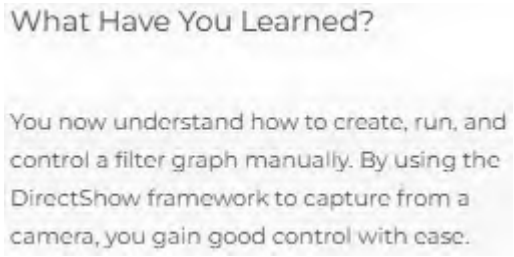
dwStart=0;

hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13<br>  <br>hResult=pMediaSeeking->GetCurrentPositio<br>hResult= pCaptureGraphBuilder->ControlSt<br>&PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid<br>&dwStart, &dwEnd, wStartCookie, wEndC<br>The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.<br>Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.<br>You now have the graph ready and running. You can start using it to capture and save in an .asf file. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1076 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See* limitations 1[a], 1[h], 1[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14b] providing instructions to the user for capturing the check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14d] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14e] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14f] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14g] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14h] before the submitting, confirming that the mobile check | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is |

| Claims | Prior Art |
|---|---|
| deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; | performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br><br>*See* limitation 10[1h].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14i] performing image processing operations on the image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "performing image processing operations on the image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device." For example:<br><br>*See* limitations 1[h], 1[k].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14j] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit, | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit." For example:<br><br>*See* limitation 1[l].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [14k] wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations comprising: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the downloaded app, when executed by the user's mobile device, is configured to cause the user's mobile device to perform additional operations." For example:<br>*See* limitation 1[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14l] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 1[n].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [14m] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 1[p].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 15[pre]. A method enabling mobile check deposit auto-capture functionality via video frame processing, the method comprising: | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] method enabling mobile check deposit auto-capture functionality via video frame processing, the method." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [15a] receiving an image of a check, wherein the image of the check is provided for mobile check deposit in a bank by a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer and a downloaded app, the downloaded app associated with the bank to control check deposit by controlling the camera through the software abstraction layer, and the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations comprising: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving an image of a check, wherein the image of the check is provided for mobile check deposit in a bank by a mobile device of a user, the user's mobile device including a camera and a memory storing a software abstraction layer and a downloaded app, the downloaded app associated with the bank to control check deposit by controlling the camera through the software abstraction layer, and the downloaded app, when executed by the user's mobile device, being configured to cause the user's mobile device to perform operations." For example:<br>*See* limitation 14[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15b] providing instructions to the user for capturing the check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing instructions to the user for capturing the check." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [15d] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15e] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15f] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15g] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[h]. |

| Claims | Prior Art |
|---|---|
| frame or is based on the captured video frame; and | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15h] before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before the submitting, confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 10[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15i] performing image processing operations on the image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "performing image processing operations on the image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device." For example:<br>*See* limitation 14[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15j] updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "updating, by a computer, a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 14[j]. |

| Claims | Prior Art |
|---|---|
| check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit, | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [15k] wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* limitation 2[m].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 16. The method of claim 15, wherein the plurality of monitoring criteria includes a criterion based on magnetic ink character recognition (MICR) line detection. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he method of claim 15, wherein the plurality of monitoring criteria includes a criterion based on magnetic ink character recognition (MICR) line detection." For example:<br>*See* claim 3.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

**Exhibit B3 - Invalidity of 12,211,095 in view of Byrne**

US 2006/0249567 A1 ("Byrne") anticipates and/or renders obvious (alone or in combination with other references), at least under USAA's apparent infringement theories, all asserted claims of U.S. Patent No. 12,211,095 ("the '095 patent"). Byrne was filed on February 9, 2006 and published on November 9, 2006 and is therefore prior art to the '095 patent.

Regions has charted the portions of the reference that correspond to each element, including at least under USAA's apparent interpretation of the claims as reflected in USAA's Complaint and Infringement Contentions. Where the chart below states that the reference "discloses," "teaches," or "suggests" a claim limitation, such disclosure, teaching, or suggestion may be express, implied, or inherent. The cited evidence is merely illustrative, and Regions reserves the right to cite alternative or additional evidence.

No part of this exemplary chart construes, or is intended to construe, the specification, file history, or claims of the asserted patents. In addition, nothing stated in this chart shall be treated as an admission or suggestion that Regions agrees with USAA regarding either the scope of any of the asserted claims or Regions' apparent interpretation of the scope of the asserted claims, or that Regions' accused products meet any limitations of the claims.

To the extent that USAA contends that Byrne does not disclose one or more limitations of the asserted claims, it would have been obvious to combine the teachings of Byrne with (1) the knowledge of one of ordinary skill in the art, and/or (2) the teachings of any of the prior art set forth in Regions' other invalidity charts and cover pleading, to show all the limitations of the asserted claims, including for the reasons set forth in Regions' Invalidity Contentions Cover Pleading, which are incorporated herein.

The cited disclosure further shows that it would have been obvious to combine Byrne with the prior art set forth in Regions' other invalidity charts and cover pleading, for example by showing similarities with the other prior art and/or elements that could be improved by combination, replacement, and/or modification in view of the other prior art.

Because USAA has yet to identify any limitation of the asserted claims that it contends is not fully disclosed by Byrne, either alone or in combination with other prior art cited by Regions, Regions expressly reserves the right to rebut any such contention, including by identifying additional § 103 combinations, if any such contention is made by USAA.

| **Claims** | **Prior Art** |
|---|---|
| 1[pre]. A system enabling mobile check deposit auto-capture functionality via video frame processing, the system comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] system enabling mobile check deposit auto-capture functionality via video frame processing, the system." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1086 of 1772

| Claims | Prior Art |
| --- | --- |
| | using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively |

| Claims | Prior Art |
| --- | --- |
| | examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized |

| Claims | Prior Art |
| --- | --- |
| | version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1089 of 1772

| Claims | Prior Art |
| --- | --- |
| | download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application; |

| Claims | Prior Art |
| --- | --- |
| | provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1091 of 1772

| Claims | Prior Art |
| --- | --- |
| | provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0044]<br>Three of the components: PC, scanner, and device drivers can be packaged by any major computer manufacturer. The application software provider can either make special arrangements with the PC provider to include the software or the provider can use other means to distribute the software. The enterprise application is the responsibility of the enterprise.<br><br>Byrne at [0045]<br>Application software can be packaged in two basic forms. The first form is normally distributed by media, such as CD-ROM, or electronically downloaded. Once in the hands of the user, the application software is installed on the machine where it stays resident on disk and is loaded into memory when used. The software industry refers this type of application as a fat application, i.e. stays on machine. Application software, installed in this manner, also normally has access to the resources such as scanners, disk drives, and other applications available on the machine within the authorization scope of the user. Additionally, software maintenance, changes or new features for fat applications require the software provider to distribute the software and for the user to install the software. If the maintenance is faulty, additional roll-back steps or software maintenance is required. Because not all users can be expected to install software at the same time, software providers are forced into a position to support multiple, often back-leveled versions of software.<br><br>Byrne at [0046]<br>The second form is distributed with each invocation of the application software through the ubiquitous browser interface available on almost every modern personal computer. The software industry refers to this type of application as a thin client. No software is installed on the disk. The application software is the same version for all users because the plug-in is downloaded with each invocation of the application software, i.e. automatic maintenance. The plug-in does not automatically have the same level of access to the resources that the user has. This security feature allows the user to give permission to the resource being requested, e.g. scanner, not all of the resources available to the user. The combined benefits of security with ease and speed of software maintenance makes the thin-client approach a more beneficial |

| Claims | Prior Art |
| --- | --- |
| | approach to both the customer and the enterprise.<br><br>Byrne at [0047]<br>One embodiment of the invention can be described with reference to FIG. 1, a schematic flow diagram showing a high level flow of a desktop deposit according to the invention. From a client means, for example, a personal computer type computer (PC), a client/user authenticates with a desktop deposit application 102. In this example, the server related software is referred to as Wells Fargo's Commercial Electronic Offices$^{SM}$ Portal (CEO Portal or just CEO) and the desktop deposit application is referred to as C21 App. After authentication, C21 App is launched and the appropriate desktop deposit plug-in(s) is downloaded onto the client PC.<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1093 of 1772

| Claims | Prior Art |
| --- | --- |
| | to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1094 of 1772

| Claims | Prior Art |
| --- | --- |
| | enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0059]<br>In this particular example of one embodiment of the invention, an ActiveX component (VC++) is used for interacting with the device API, referred to herein as Ranger API, for scanning the checks. The Envoy client is used for retrieving the entitlements. The Envoy client uses T3 protocol, which is proprietary to WebLogic Application Server. DD Application Server uses HTTPS/Post for interfacing with AIM. A Wells Fargo standard, WFXML, is followed for making the request/response to AIM service. DD Application Server posts messages in X9.37 format to ICING MQSeries queue. DD Application Server uses standard JDBC implementation to interact with Work In Progress (local) Database. For retrieving Account Name needs to interact with internal system using Xstream Interface. For RTN validation an external system is used. It should be appreciated that the implementation described hereinabove is for illustrative purposes and is not meant to be limiting.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1095 of 1772

| Claims | Prior Art |
| --- | --- |
| | to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1097 of 1772

| Claims | Prior Art |
| --- | --- |
| | The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0142]–[0144]<br>Automated Updates.<br>The CLASP component 406 is designed as a signed ActiveX control. The control is deployed as a cab, i.e. Windows cabinet, file. This allows the control to be downloaded automatically whenever a new version of the control is posted to the CEO web server.<br>When the CEO application page references a new control version, the user's browser 408 automatically downloads or prompts for download of the control depending upon the browser security settings.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0149]–[0150]<br>Browser and OS.<br>By way of example, in one embodiment of the invention, the CLASP component is designed to work with the IE 5.01 (SP2) or IE 6 browsers. The operating systems supported are Windows XP and Windows 2000.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in |

| Claims | Prior Art |
| --- | --- |
| | JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1100 of 1772

| Claims | Prior Art |
| --- | --- |
| | scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1101 of 1772

| Claims | Prior Art |
| --- | --- |
| | ActiveX components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 8<br>8. The computer implemented method of claim 1, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 10<br>10. The computer implemented method of claim 1, further comprising the step of:<br>responsive to a new version of said one or more client components posted to said enterprise server component, said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1102 of 1772

| Claims | Prior Art |
| --- | --- |
| | regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 25<br>25. The system of claim 16, further comprising:<br>responsive to a new version of said one or more client components posted to said enterprise server |

| Claims | Prior Art |
|---|---|
| | component, means for said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1a] a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing." For example:<br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and |

| Claims | Prior Art |
| --- | --- |
| | present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image |

| Claims | Prior Art |
| --- | --- |
| | processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a |

| Claims | Prior Art |
| --- | --- |
| | paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1108 of 1772

| Claims | Prior Art |
| --- | --- |
| | of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks |

| Claims | Prior Art |
| --- | --- |
| | using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0044]<br>Three of the components: PC, scanner, and device drivers can be packaged by any major computer manufacturer. The application software provider can either make special arrangements with the PC provider to include the software or the provider can use other means to distribute the software. The enterprise application is the responsibility of the enterprise. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0045]<br>Application software can be packaged in two basic forms. The first form is normally distributed by media, such as CD-ROM, or electronically downloaded. Once in the hands of the user, the application software is installed on the machine where it stays resident on disk and is loaded into memory when used. The software industry refers this type of application as a fat application, i.e. stays on machine. Application software, installed in this manner, also normally has access to the resources such as scanners, disk drives, and other applications available on the machine within the authorization scope of the user. Additionally, software maintenance, changes or new features for fat applications require the software provider to distribute the software and for the user to install the software. If the maintenance is faulty, additional roll-back steps or software maintenance is required. Because not all users can be expected to install software at the same time, software providers are forced into a position to support multiple, often back-leveled versions of software.<br><br>Byrne at [0046]<br>The second form is distributed with each invocation of the application software through the ubiquitous browser interface available on almost every modern personal computer. The software industry refers to this type of application as a thin client. No software is installed on the disk. The application software is the same version for all users because the plug-in is downloaded with each invocation of the application software, i.e. automatic maintenance. The plug-in does not automatically have the same level of access to the resources that the user has. This security feature allows the user to give permission to the resource being requested, e.g. scanner, not all of the resources available to the user. The combined benefits of security with ease and speed of software maintenance makes the thin-client approach a more beneficial approach to both the customer and the enterprise.<br><br>Byrne at [0047]<br>One embodiment of the invention can be described with reference to FIG. 1, a schematic flow diagram showing a high level flow of a desktop deposit according to the invention. From a client means, for example, a personal computer type computer (PC), a client/user authenticates with a desktop deposit application 102. In this example, the server related software is referred to as Wells Fargo's Commercial Electronic Offices[SM] Portal (CEO Portal or just CEO) and the desktop deposit application is referred to as C21 App. After authentication, C21 App is launched and the appropriate desktop deposit plug-in(s) is downloaded onto the client PC. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1111 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0059]<br>In this particular example of one embodiment of the invention, an ActiveX component (VC++) is used for interacting with the device API, referred to herein as Ranger API, for scanning the checks. The Envoy client is used for retrieving the entitlements. The Envoy client uses T3 protocol, which is proprietary to WebLogic Application Server. DD Application Server uses HTTPS/Post for interfacing with AIM. A Wells Fargo standard, WFXML, is followed for making the request/response to AIM service. DD Application Server posts messages in X9.37 format to ICING MQSeries queue. DD Application Server uses standard JDBC implementation to interact with Work In Progress (local) Database. For retrieving Account Name needs to interact with internal system using Xstream Interface. For RTN validation an |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1113 of 1772

| Claims | Prior Art |
| --- | --- |
| | external system is used. It should be appreciated that the implementation described hereinabove is for illustrative purposes and is not meant to be limiting.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1115 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0142]–[0144]<br>Automated Updates.<br>The CLASP component 406 is designed as a signed ActiveX control. The control is deployed as a cab, i.e. Windows cabinet, file. This allows the control to be downloaded automatically whenever a new version of the control is posted to the CEO web server.<br>When the CEO application page references a new control version, the user's browser 408 automatically downloads or prompts for download of the control depending upon the browser security settings.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0149]–[0150]<br>Browser and OS.<br>By way of example, in one embodiment of the invention, the CLASP component is designed to work with the IE 5.01 (SP2) or IE 6 browsers. The operating systems supported are Windows XP and Windows 2000.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1116 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in |

| Claims | Prior Art |
| --- | --- |
| | JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1118 of 1772

| Claims | Prior Art |
| --- | --- |
| | scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1119 of 1772

| Claims | Prior Art |
| --- | --- |
| | ActiveX components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 8<br>8. The computer implemented method of claim 1, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 10<br>10. The computer implemented method of claim 1, further comprising the step of:<br>responsive to a new version of said one or more client components posted to said enterprise server component, said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1120 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1121 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 25<br>25. The system of claim 16, further comprising:<br>responsive to a new version of said one or more client components posted to said enterprise server component, means for said browser initiating a download of said one or more client components of the new version. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1122 of 1772

| Claims | Prior Art |
|---|---|
| | Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitation 1[pre].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1b] a software abstraction layer for controlling the camera, the software abstraction layer comprising Instructions that, when executed by the processor, control the camera; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a software abstraction layer for controlling the camera, the software abstraction layer comprising Instructions that, when executed by the processor, control the camera." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting |

| Claims | Prior Art |
| --- | --- |
| | checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where |

| Claims | Prior Art |
| --- | --- |
| | images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the |

| Claims | Prior Art |
| --- | --- |
| | RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1127 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0044]<br>Three of the components: PC, scanner, and device drivers can be packaged by any major computer manufacturer. The application software provider can either make special arrangements with the PC provider to include the software or the provider can use other means to distribute the software. The enterprise application is the responsibility of the enterprise.<br><br>Byrne at [0045]<br>Application software can be packaged in two basic forms. The first form is normally distributed by media, such as CD-ROM, or electronically downloaded. Once in the hands of the user, the application |

| Claims | Prior Art |
| --- | --- |
| | software is installed on the machine where it stays resident on disk and is loaded into memory when used. The software industry refers this type of application as a fat application, i.e. stays on machine. Application software, installed in this manner, also normally has access to the resources such as scanners, disk drives, and other applications available on the machine within the authorization scope of the user. Additionally, software maintenance, changes or new features for fat applications require the software provider to distribute the software and for the user to install the software. If the maintenance is faulty, additional roll-back steps or software maintenance is required. Because not all users can be expected to install software at the same time, software providers are forced into a position to support multiple, often back-leveled versions of software.<br><br>Byrne at [0046]<br>The second form is distributed with each invocation of the application software through the ubiquitous browser interface available on almost every modern personal computer. The software industry refers to this type of application as a thin client. No software is installed on the disk. The application software is the same version for all users because the plug-in is downloaded with each invocation of the application software, i.e. automatic maintenance. The plug-in does not automatically have the same level of access to the resources that the user has. This security feature allows the user to give permission to the resource being requested, e.g. scanner, not all of the resources available to the user. The combined benefits of security with ease and speed of software maintenance makes the thin-client approach a more beneficial approach to both the customer and the enterprise.<br><br>Byrne at [0047]<br>One embodiment of the invention can be described with reference to FIG. 1, a schematic flow diagram showing a high level flow of a desktop deposit according to the invention. From a client means, for example, a personal computer type computer (PC), a client/user authenticates with a desktop deposit application 102. In this example, the server related software is referred to as Wells Fargo's Commercial Electronic Offices[SM] Portal (CEO Portal or just CEO) and the desktop deposit application is referred to as C21 App. After authentication, C21 App is launched and the appropriate desktop deposit plug-in(s) is downloaded onto the client PC.<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1129 of 1772

| Claims | Prior Art |
| --- | --- |
| | called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is |

| Claims | Prior Art |
| --- | --- |
| | authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0059]<br>In this particular example of one embodiment of the invention, an ActiveX component (VC++) is used for interacting with the device API, referred to herein as Ranger API, for scanning the checks. The Envoy client is used for retrieving the entitlements. The Envoy client uses T3 protocol, which is proprietary to WebLogic Application Server. DD Application Server uses HTTPS/Post for interfacing with AIM. A Wells Fargo standard, WFXML, is followed for making the request/response to AIM service. DD Application Server posts messages in X9.37 format to ICING MQSeries queue. DD Application Server uses standard JDBC implementation to interact with Work In Progress (local) Database. For retrieving Account Name needs to interact with internal system using Xstream Interface. For RTN validation an external system is used. It should be appreciated that the implementation described hereinabove is for |

| Claims | Prior Art |
| --- | --- |
| | illustrative purposes and is not meant to be limiting.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following: |

| Claims | Prior Art |
| --- | --- |
| | How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0142]–[0144]<br>Automated Updates.<br>The CLASP component 406 is designed as a signed ActiveX control. The control is deployed as a cab, i.e. Windows cabinet, file. This allows the control to be downloaded automatically whenever a new |

| Claims | Prior Art |
| --- | --- |
| | version of the control is posted to the CEO web server.<br>When the CEO application page references a new control version, the user's browser 408 automatically downloads or prompts for download of the control depending upon the browser security settings.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0149]–[0150]<br>Browser and OS.<br>By way of example, in one embodiment of the invention, the CLASP component is designed to work with the IE 5.01 (SP2) or IE 6 browsers. The operating systems supported are Windows XP and Windows 2000.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the |

| Claims | Prior Art |
| --- | --- |
| | images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1137 of 1772

| Claims | Prior Art |
| --- | --- |
| | responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a |

| Claims | Prior Art |
| --- | --- |
| | variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 8<br>8. The computer implemented method of claim 1, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 10<br>10. The computer implemented method of claim 1, further comprising the step of:<br>responsive to a new version of said one or more client components posted to said enterprise server component, said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising: |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1139 of 1772

| Claims | Prior Art |
| --- | --- |
| | means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1140 of 1772

| Claims | Prior Art |
|---|---|
| | Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 25<br>25. The system of claim 16, further comprising:<br>responsive to a new version of said one or more client components posted to said enterprise server component, means for said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component |

| Claims | Prior Art |
|---|---|
| | as an HTTP message.<br><br>*See also* limitations 1[pre], 1[a].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1c] a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, the downloaded app comprising Instructions that, when executed by the processor, cause the user's mobile device to perform operations including: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, the downloaded app comprising Instructions that, when executed by the processor, cause the user's mobile device to perform operations including." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for |

| Claims | Prior Art |
| --- | --- |
| | allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an |

| Claims | Prior Art |
| --- | --- |
| | image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of |

| Claims | Prior Art |
| --- | --- |
| | requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method |

| Claims | Prior Art |
| --- | --- |
| | for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0044]<br>Three of the components: PC, scanner, and device drivers can be packaged by any major computer manufacturer. The application software provider can either make special arrangements with the PC provider to include the software or the provider can use other means to distribute the software. The enterprise application is the responsibility of the enterprise.<br><br>Byrne at [0045]<br>Application software can be packaged in two basic forms. The first form is normally distributed by media, such as CD-ROM, or electronically downloaded. Once in the hands of the user, the application software is installed on the machine where it stays resident on disk and is loaded into memory when used. The software industry refers this type of application as a fat application, i.e. stays on machine. Application software, installed in this manner, also normally has access to the resources such as scanners, disk drives, and other applications available on the machine within the authorization scope of the user. Additionally, |

| Claims | Prior Art |
| --- | --- |
| | software maintenance, changes or new features for fat applications require the software provider to distribute the software and for the user to install the software. If the maintenance is faulty, additional roll-back steps or software maintenance is required. Because not all users can be expected to install software at the same time, software providers are forced into a position to support multiple, often back-leveled versions of software.<br><br>Byrne at [0046]<br>The second form is distributed with each invocation of the application software through the ubiquitous browser interface available on almost every modern personal computer. The software industry refers to this type of application as a thin client. No software is installed on the disk. The application software is the same version for all users because the plug-in is downloaded with each invocation of the application software, i.e. automatic maintenance. The plug-in does not automatically have the same level of access to the resources that the user has. This security feature allows the user to give permission to the resource being requested, e.g. scanner, not all of the resources available to the user. The combined benefits of security with ease and speed of software maintenance makes the thin-client approach a more beneficial approach to both the customer and the enterprise.<br><br>Byrne at [0047]<br>One embodiment of the invention can be described with reference to FIG. 1, a schematic flow diagram showing a high level flow of a desktop deposit according to the invention. From a client means, for example, a personal computer type computer (PC), a client/user authenticates with a desktop deposit application 102. In this example, the server related software is referred to as Wells Fargo's Commercial Electronic Offices[SM] Portal (CEO Portal or just CEO) and the desktop deposit application is referred to as C21 App. After authentication, C21 App is launched and the appropriate desktop deposit plug-in(s) is downloaded onto the client PC.<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been |

| Claims | Prior Art |
| --- | --- |
| | electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. |

| Claims | Prior Art |
| --- | --- |
| | The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0059]<br>In this particular example of one embodiment of the invention, an ActiveX component (VC++) is used for interacting with the device API, referred to herein as Ranger API, for scanning the checks. The Envoy client is used for retrieving the entitlements. The Envoy client uses T3 protocol, which is proprietary to WebLogic Application Server. DD Application Server uses HTTPS/Post for interfacing with AIM. A Wells Fargo standard, WFXML, is followed for making the request/response to AIM service. DD Application Server posts messages in X9.37 format to ICING MQSeries queue. DD Application Server uses standard JDBC implementation to interact with Work In Progress (local) Database. For retrieving Account Name needs to interact with internal system using Xstream Interface. For RTN validation an external system is used. It should be appreciated that the implementation described hereinabove is for illustrative purposes and is not meant to be limiting.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer |

| Claims | Prior Art |
| --- | --- |
| | profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB |

| Claims | Prior Art |
| --- | --- |
| | information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1152 of 1772

| Claims | Prior Art |
| --- | --- |
| | What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0142]–[0144]<br>Automated Updates.<br>The CLASP component 406 is designed as a signed ActiveX control. The control is deployed as a cab, i.e. Windows cabinet, file. This allows the control to be downloaded automatically whenever a new version of the control is posted to the CEO web server.<br>When the CEO application page references a new control version, the user's browser 408 automatically downloads or prompts for download of the control depending upon the browser security settings. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0149]–[0150]<br>Browser and OS.<br>By way of example, in one embodiment of the invention, the CLASP component is designed to work with the IE 5.01 (SP2) or IE 6 browsers. The operating systems supported are Windows XP and Windows 2000.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses |

| Claims | Prior Art |
| --- | --- |
| | a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device; |

| Claims | Prior Art |
| --- | --- |
| | from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 8<br>8. The computer implemented method of claim 1, wherein said one or more client components provide an |

| Claims | Prior Art |
| --- | --- |
| | interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 10<br>10. The computer implemented method of claim 1, further comprising the step of:<br>responsive to a new version of said one or more client components posted to said enterprise server component, said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1158 of 1772

| Claims | Prior Art |
| --- | --- |
| | device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1159 of 1772

| Claims | Prior Art |
|---|---|
| | controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 25<br>25. The system of claim 16, further comprising:<br>responsive to a new version of said one or more client components posted to said enterprise server component, means for said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[pre]–1[b].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [1d] providing Instructions to the user for capturing an image of a check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing Instructions to the user for capturing an image of a check." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for |

| Claims | Prior Art |
| --- | --- |
| | Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of |

| Claims | Prior Art |
| --- | --- |
| | paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be |

| Claims | Prior Art |
| --- | --- |
| | processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1164 of 1772

| Claims | Prior Art |
| --- | --- |
| | provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any |

| Claims | Prior Art |
| --- | --- |
| | digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability |

| Claims | Prior Art |
| --- | --- |
| | to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1167 of 1772

| Claims | Prior Art |
| --- | --- |
| | enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and |

| Claims | Prior Art |
| --- | --- |
| | What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses |

| Claims | Prior Art |
| --- | --- |
| | JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1170 of 1772

| Claims | Prior Art |
| --- | --- |
| | posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1171 of 1772

| Claims | Prior Art |
| --- | --- |
| | deposit is a bank check, coupon, or document.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more |

| Claims | Prior Art |
|---|---|
| | captured objects is one or more scanned images.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>*See also* limitation 1[pre].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1e] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known |

| Claims | Prior Art |
| --- | --- |
| | problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a |

| Claims | Prior Art |
| --- | --- |
| | personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing |

| Claims | Prior Art |
| --- | --- |
| | environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0044]<br>Three of the components: PC, scanner, and device drivers can be packaged by any major computer manufacturer. The application software provider can either make special arrangements with the PC provider to include the software or the provider can use other means to distribute the software. The enterprise application is the responsibility of the enterprise.<br><br>Byrne at [0045]<br>Application software can be packaged in two basic forms. The first form is normally distributed by media, such as CD-ROM, or electronically downloaded. Once in the hands of the user, the application software is installed on the machine where it stays resident on disk and is loaded into memory when used. The software industry refers this type of application as a fat application, i.e. stays on machine. Application software, installed in this manner, also normally has access to the resources such as scanners, disk drives, |

| Claims | Prior Art |
| --- | --- |
| | and other applications available on the machine within the authorization scope of the user. Additionally, software maintenance, changes or new features for fat applications require the software provider to distribute the software and for the user to install the software. If the maintenance is faulty, additional roll-back steps or software maintenance is required. Because not all users can be expected to install software at the same time, software providers are forced into a position to support multiple, often back-leveled versions of software.<br><br>Byrne at [0046]<br>The second form is distributed with each invocation of the application software through the ubiquitous browser interface available on almost every modern personal computer. The software industry refers to this type of application as a thin client. No software is installed on the disk. The application software is the same version for all users because the plug-in is downloaded with each invocation of the application software, i.e. automatic maintenance. The plug-in does not automatically have the same level of access to the resources that the user has. This security feature allows the user to give permission to the resource being requested, e.g. scanner, not all of the resources available to the user. The combined benefits of security with ease and speed of software maintenance makes the thin-client approach a more beneficial approach to both the customer and the enterprise.<br><br>Byrne at [0047]<br>One embodiment of the invention can be described with reference to FIG. 1, a schematic flow diagram showing a high level flow of a desktop deposit according to the invention. From a client means, for example, a personal computer type computer (PC), a client/user authenticates with a desktop deposit application 102. In this example, the server related software is referred to as Wells Fargo's Commercial Electronic Offices[SM] Portal (CEO Portal or just CEO) and the desktop deposit application is referred to as C21 App. After authentication, C21 App is launched and the appropriate desktop deposit plug-in(s) is downloaded onto the client PC.<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can |

| Claims | Prior Art |
| --- | --- |
| | be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can |

| Claims | Prior Art |
| --- | --- |
| | provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0059]<br>In this particular example of one embodiment of the invention, an ActiveX component (VC++) is used for interacting with the device API, referred to herein as Ranger API, for scanning the checks. The Envoy client is used for retrieving the entitlements. The Envoy client uses T3 protocol, which is proprietary to WebLogic Application Server. DD Application Server uses HTTPS/Post for interfacing with AIM. A Wells Fargo standard, WFXML, is followed for making the request/response to AIM service. DD Application Server posts messages in X9.37 format to ICING MQSeries queue. DD Application Server uses standard JDBC implementation to interact with Work In Progress (local) Database. For retrieving Account Name needs to interact with internal system using Xstream Interface. For RTN validation an external system is used. It should be appreciated that the implementation described hereinabove is for illustrative purposes and is not meant to be limiting.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1180 of 1772

| Claims | Prior Art |
| --- | --- |
| | component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0142]–[0144]<br>Automated Updates.<br>The CLASP component 406 is designed as a signed ActiveX control. The control is deployed as a cab, i.e. Windows cabinet, file. This allows the control to be downloaded automatically whenever a new version of the control is posted to the CEO web server. |

| Claims | Prior Art |
| --- | --- |
| | When the CEO application page references a new control version, the user's browser 408 automatically downloads or prompts for download of the control depending upon the browser security settings.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0149]–[0150]<br>Browser and OS.<br>By way of example, in one embodiment of the invention, the CLASP component is designed to work with the IE 5.01 (SP2) or IE 6 browsers. The operating systems supported are Windows XP and Windows 2000.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one |

| Claims | Prior Art |
| --- | --- |
| | embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the |

| Claims | Prior Art |
| --- | --- |
| | MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device; |

| Claims | Prior Art |
| --- | --- |
| | from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 8<br>8. The computer implemented method of claim 1, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1185 of 1772

| Claims | Prior Art |
| --- | --- |
| | more different capturing device types.<br><br>Byrne at claim 10<br>10. The computer implemented method of claim 1, further comprising the step of:<br>responsive to a new version of said one or more client components posted to said enterprise server component, said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted |

| Claims | Prior Art |
| --- | --- |
| | additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 25<br>25. The system of claim 16, further comprising:<br>responsive to a new version of said one or more client components posted to said enterprise server |

| Claims | Prior Art |
|---|---|
| | component, means for said browser initiating a download of said one or more client components of the new version.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[pre]–1[c].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1f] overlaying an alignment guide on the live video of the check in the field of view of the camera; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors |

| Claims | Prior Art |
| --- | --- |
| | which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner |

| Claims | Prior Art |
| --- | --- |
| | model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1191 of 1772

| Claims | Prior Art |
| --- | --- |
| | images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and |

| Claims | Prior Art |
| --- | --- |
| | said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1195 of 1772

| Claims | Prior Art |
|---|---|
| | device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[pre]–1[b].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1g] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example: |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1197 of 1772

| Claims | Prior Art |
| --- | --- |
| | system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism |

| Claims | Prior Art |
| --- | --- |
| | for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of: |

| Claims | Prior Art |
| --- | --- |
| | one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are |

| Claims | Prior Art |
| --- | --- |
| | developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server; |

| Claims | Prior Art |
| --- | --- |
| | means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted |

| Claims | Prior Art |
|---|---|
| | information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[pre]–1[b], 1[e].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1h] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can |

| Claims | Prior Art |
| --- | --- |
| | download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1207 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form- |

| Claims | Prior Art |
| --- | --- |
| | post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to |

| Claims | Prior Art |
| --- | --- |
| | indicate that the check had been properly scanned.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary |

| Claims | Prior Art |
| --- | --- |
| | information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising: |

| Claims | Prior Art |
| --- | --- |
| | means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents. |

| Claims | Prior Art |
|---|---|
| | Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[pre]–1[b], 1[e], 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1i] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals: |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1213 of 1772

| Claims | Prior Art |
| --- | --- |
| user that the check has been captured; | provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug- |

| Claims | Prior Art |
| --- | --- |
| | in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and |

| Claims | Prior Art |
| --- | --- |
| | to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following: |

| Claims | Prior Art |
| --- | --- |
| | How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component |

| Claims | Prior Art |
| --- | --- |
| | as an in-process server.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1218 of 1772

| Claims | Prior Art |
| --- | --- |
| | access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is |

| Claims | Prior Art |
| --- | --- |
| | posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1220 of 1772

| Claims | Prior Art |
| --- | --- |
| | ActiveX components.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1221 of 1772

| Claims | Prior Art |
| --- | --- |
| | for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that |

| Claims | Prior Art |
|---|---|
| | said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[pre]–1[b], 1[e], 1[g], 1[h].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1j] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates |

| Claims | Prior Art |
| --- | --- |
| | to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing |

| Claims | Prior Art |
| --- | --- |
| | with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital |

| Claims | Prior Art |
| --- | --- |
| | signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed |

| Claims | Prior Art |
| --- | --- |
| | machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0014]<br>It would be advantageous to improve operating efficiencies by providing faster access to money deposited in a customer's account and providing increased flexibility to invest and direct funds.<br><br>Byrne at [0015]<br>It would further be advantageous to provide a system and method that can handle reductions in float as the time to settle transactions between banks decreases.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical |

| Claims | Prior Art |
| --- | --- |
| | goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following: |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1229 of 1772

| Claims | Prior Art |
| --- | --- |
| | scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series. |

| Claims | Prior Art |
| --- | --- |
| | The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0088]<br>This file format is based on the ANSI X9.37 DSTU—2003 "Specifications for Electronic Exchange of Check and Image Data" draft standard with some modifications to accommodate the processing of credits, such as customer Direct Deposit Account (DDA) deposits and adjustments, and to augment sender source identification and file balancing controls.<br><br>Byrne at [0089]<br>To ensure proper processing, it is important that all deposits on a file are transmitted in sequential order, and, for example, deposit credit before associated checks, and that each credit and debit item is assigned a unique, i.e. within business day, Item Sequence Number (ISN).<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user: |

| Claims | Prior Art |
|---|---|
| | Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types.<br>Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future. |

| Claims | Prior Art |
| --- | --- |
| | Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. |

| Claims | Prior Art |
| --- | --- |
| | In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to |

| Claims | Prior Art |
| --- | --- |
| | indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0185]<br>The CEO system and method then takes the incoming request and access the current session deposit bag 622. It takes the session ID and tags it to each item, having identified the item sequence number.<br><br>Byrne at [0186]<br>The CEO system and method checks for any additional validations to be performed 624.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1236 of 1772

| Claims | Prior Art |
| --- | --- |
| | from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 11<br>11. The computer implemented method of claim 1, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 13<br>13. The computer implemented method of claim 2, further comprising the step of:<br>providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 14<br>14. The computer implemented method of claim 1, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in |

| Claims | Prior Art |
| --- | --- |
| | approach is implemented.<br><br>Byrne at claim 15<br>15. The computer implemented method of claim 1, wherein any of said captured one or more objects is replaced by a file upload.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device.<br><br>Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 26<br>The system of claim 16, wherein said one or more client components interact through HTTP messages. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1240 of 1772

| Claims | Prior Art |
|---|---|
| | Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 28<br>28. The system of claim 17, further comprising:<br>means for providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 29<br>29. The system of claim 16, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented.<br><br>Byrne at claim 30<br>30. The system of claim 16, wherein any of said captured one or more objects is replaced by a file upload.<br><br>*See also* limitations 1[pre], 1[e], 1[i].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1k] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks |

| Claims | Prior Art |
| --- | --- |
| on the check in the digital image is validated; | using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively |

| Claims | Prior Art |
| --- | --- |
| | examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized |

| Claims | Prior Art |
| --- | --- |
| | version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed |

| Claims | Prior Art |
| --- | --- |
| | download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application; |

| Claims | Prior Art |
| --- | --- |
| | provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers |

| Claims | Prior Art |
| --- | --- |
| | provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation; |

| Claims | Prior Art |
| --- | --- |
| | single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance |

| Claims | Prior Art |
| --- | --- |
| | purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One |

| Claims | Prior Art |
| --- | --- |
| | embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0167]<br>Deposit bag is a holding/receiving area until the customer commits the deposit. Items are not committed to ICING until the bag is closed;<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an |

| Claims | Prior Art |
| --- | --- |
| | acknowledgment for each scanned item; and<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0185]<br>The CEO system and method then takes the incoming request and access the current session deposit bag 622. It takes the session ID and tags it to each item, having identified the item sequence number.<br><br>Byrne at [0186]<br>The CEO system and method checks for any additional validations to be performed 624.<br><br>Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device; |

| Claims | Prior Art |
| --- | --- |
| | from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 4<br>4. The computer implemented method of claim 1, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 13<br>13. The computer implemented method of claim 2, further comprising the step of:<br>providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1253 of 1772

| Claims | Prior Art |
| --- | --- |
| | components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 19<br>19. The system of claim 16, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, |

| Claims | Prior Art |
|---|---|
| | coupon, or document.<br><br>Byrne at claim 28<br>28. The system of claim 17, further comprising:<br>means for providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>*See also* limitation 1[pre].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1l] a server, remote from the user's mobile device, comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the server to perform operations including: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a server, remote from the user's mobile device, comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the server to perform operations including." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks |

| Claims | Prior Art |
| --- | --- |
| | from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal |

| Claims | Prior Art |
| --- | --- |
| | (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0014]<br>It would be advantageous to improve operating efficiencies by providing faster access to money deposited in a customer's account and providing increased flexibility to invest and direct funds.<br><br>Byrne at [0015]<br>It would further be advantageous to provide a system and method that can handle reductions in float as the time to settle transactions between banks decreases.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML |

| Claims | Prior Art |
| --- | --- |
| | service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0088]<br>This file format is based on the ANSI X9.37 DSTU—2003 "Specifications for Electronic Exchange of Check and Image Data" draft standard with some modifications to accommodate the processing of credits, such as customer Direct Deposit Account (DDA) deposits and adjustments, and to augment sender source identification and file balancing controls. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0089]<br>To ensure proper processing, it is important that all deposits on a file are transmitted in sequential order, and, for example, deposit credit before associated checks, and that each credit and debit item is assigned a unique, i.e. within business day, Item Sequence Number (ISN).<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0185]<br>The CEO system and method then takes the incoming request and access the current session deposit bag 622. It takes the session ID and tags it to each item, having identified the item sequence number.<br><br>Byrne at [0186]<br>The CEO system and method checks for any additional validations to be performed 624.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user |

| Claims | Prior Art |
| --- | --- |
| | with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 11<br>11. The computer implemented method of claim 1, wherein said one or more client components interact through HTTP messages. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1261 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 13<br>13. The computer implemented method of claim 2, further comprising the step of:<br>providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 14<br>14. The computer implemented method of claim 1, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented.<br><br>Byrne at claim 15<br>15. The computer implemented method of claim 1, wherein any of said captured one or more objects is replaced by a file upload.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate |

| Claims | Prior Art |
| --- | --- |
| | with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 26<br>The system of claim 16, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 28<br>28. The system of claim 17, further comprising:<br>means for providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 29<br>29. The system of claim 16, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented. |

| Claims | Prior Art |
|---|---|
| | Byrne at claim 30<br>30. The system of claim 16, wherein any of said captured one or more objects is replaced by a file upload.<br><br>*See also* limitation 1[pre].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1m] receiving the digital image of the check transmitted by the user's mobile device, and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving the digital image of the check transmitted by the user's mobile device." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the |

| Claims | Prior Art |
| --- | --- |
| | customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and |

| Claims | Prior Art |
| --- | --- |
| | inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. |

| Claims | Prior Art |
| --- | --- |
| | O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1267 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method |

| Claims | Prior Art |
| --- | --- |
| | has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are |

| Claims | Prior Art |
| --- | --- |
| | sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size; |

| Claims | Prior Art |
| --- | --- |
| | What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in |

| Claims | Prior Art |
| --- | --- |
| | JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs |

| Claims | Prior Art |
| --- | --- |
| | immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0185]<br>The CEO system and method then takes the incoming request and access the current session deposit bag 622. It takes the session ID and tags it to each item, having identified the item sequence number.<br><br>Byrne at [0186]<br>The CEO system and method checks for any additional validations to be performed 624.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object; |

| Claims | Prior Art |
| --- | --- |
| | said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 11<br>11. The computer implemented method of claim 1, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 13<br>13. The computer implemented method of claim 2, further comprising the step of:<br>providing a deposit bag that serves as a holding area until a deposit is committed.<br>Byrne at claim 14<br>14. The computer implemented method of claim 1, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1275 of 1772

| Claims | Prior Art |
| --- | --- |
| | approach is implemented.<br><br>Byrne at claim 15<br>15. The computer implemented method of claim 1, wherein any of said captured one or more objects is replaced by a file upload.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1276 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 26<br>The system of claim 16, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 28<br>28. The system of claim 17, further comprising:<br>means for providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 29<br>29. The system of claim 16, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented.<br><br>Byrne at claim 30<br>30. The system of claim 16, wherein any of said captured one or more objects is replaced by a file upload.<br><br>*See also* limitations 1[pre], 1[j], 1[l]. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1277 of 1772

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1n] processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, the user's mobile device receiving an Indication of a result of the server's processing before confirming that the mobile check deposit can go forward; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, the user's mobile device receiving an Indication of a result of the server's processing before confirming that the mobile check deposit can go forward." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known |

| Claims | Prior Art |
| --- | --- |
| | problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted |

| Claims | Prior Art |
| --- | --- |
| | between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a |

| Claims | Prior Art |
| --- | --- |
| | Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method |

| Claims | Prior Art |
| --- | --- |
| | has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are |

| Claims | Prior Art |
| --- | --- |
| | sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size; |

| Claims | Prior Art |
| --- | --- |
| | What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in |

| Claims | Prior Art |
| --- | --- |
| | JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0167]<br>Deposit bag is a holding/receiving area until the customer commits the deposit. Items are not committed to ICING until the bag is closed;<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item; and<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, |

| Claims | Prior Art |
| --- | --- |
| | amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0185]<br>The CEO system and method then takes the incoming request and access the current session deposit bag 622. It takes the session ID and tags it to each item, having identified the item sequence number.<br><br>Byrne at [0186]<br>The CEO system and method checks for any additional validations to be performed 624.<br><br>Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1288 of 1772

| Claims | Prior Art |
| --- | --- |
| | from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 4<br>4. The computer implemented method of claim 1, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 13<br>13. The computer implemented method of claim 2, further comprising the step of:<br>providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1290 of 1772

| Claims | Prior Art |
|---|---|
| | Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 19<br>19. The system of claim 16, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 28<br>28. The system of claim 17, further comprising:<br>means for providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>*See also* limitations 1[pre], 1[k], 1[l].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1o] a computer comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the computer to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a computer comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the computer to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks |

| Claims | Prior Art |
| --- | --- |
| the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit. | using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively |

| Claims | Prior Art |
| --- | --- |
| | examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized |

| Claims | Prior Art |
| --- | --- |
| | version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed |

| Claims | Prior Art |
| --- | --- |
| | download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0014]<br>It would be advantageous to improve operating efficiencies by providing faster access to money deposited in a customer's account and providing increased flexibility to invest and direct funds.<br><br>Byrne at [0015]<br>It would further be advantageous to provide a system and method that can handle reductions in float as the time to settle transactions between banks decreases.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1295 of 1772

| Claims | Prior Art |
| --- | --- |
| | provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any |

| Claims | Prior Art |
| --- | --- |
| | type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability |

| Claims | Prior Art |
| --- | --- |
| | to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to |

| Claims | Prior Art |
| --- | --- |
| | enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0088]<br>This file format is based on the ANSI X9.37 DSTU—2003 "Specifications for Electronic Exchange of Check and Image Data" draft standard with some modifications to accommodate the processing of credits, such as customer Direct Deposit Account (DDA) deposits and adjustments, and to augment sender source identification and file balancing controls.<br><br>Byrne at [0089]<br>To ensure proper processing, it is important that all deposits on a file are transmitted in sequential order, and, for example, deposit credit before associated checks, and that each credit and debit item is assigned a unique, i.e. within business day, Item Sequence Number (ISN).<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1299 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), |

| Claims | Prior Art |
| --- | --- |
| | invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0167]<br>Deposit bag is a holding/receiving area until the customer commits the deposit. Items are not committed to ICING until the bag is closed;<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item; and<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the |

| Claims | Prior Art |
| --- | --- |
| | MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0185]<br>The CEO system and method then takes the incoming request and access the current session deposit bag 622. It takes the session ID and tags it to each item, having identified the item sequence number.<br><br>Byrne at [0186]<br>The CEO system and method checks for any additional validations to be performed 624.<br><br>Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 4<br>4. The computer implemented method of claim 1, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 11<br>11. The computer implemented method of claim 1, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 13<br>13. The computer implemented method of claim 2, further comprising the step of:<br>providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 14<br>14. The computer implemented method of claim 1, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented.<br><br>Byrne at claim 15<br>15. The computer implemented method of claim 1, wherein any of said captured one or more objects is |

| Claims | Prior Art |
| --- | --- |
| | replaced by a file upload.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more |

| Claims | Prior Art |
| --- | --- |
| | captured objects is one or more scanned images.<br><br>Byrne at claim 19<br>19. The system of claim 16, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 26<br>The system of claim 16, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 28<br>28. The system of claim 17, further comprising:<br>means for providing a deposit bag that serves as a holding area until a deposit is committed.<br>Byrne at claim 29<br>29. The system of claim 16, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented.<br><br>Byrne at claim 30<br>30. The system of claim 16, wherein any of said captured one or more objects is replaced by a file upload.<br><br>*See also* limitations 1[pre], 1[kj]–1[l], 1[n]. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1306 of 1772

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 2[pre]. The system of claim 1, wherein the downloaded app causes the user's mobile device to perform additional operations including: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 1, wherein the downloaded app causes the user's mobile device to perform additional operations including." For example:<br>*See* limitations 1[pre], 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2a] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitations 1[pre]-1[b], 1[e], 1[g], 1[h], 2[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2b] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitations 1[pre]-1[b], 1[e], 1[g]-1[i], 2[pre], 2[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 3. The system of claim 1, wherein the at least one monitoring criterion is a plurality of | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 1, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, |

| Claims | Prior Art |
| --- | --- |
| monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for |

| Claims | Prior Art |
| --- | --- |
| | Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of |

| Claims | Prior Art |
| --- | --- |
| | paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be |

| Claims | Prior Art |
| --- | --- |
| | processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1311 of 1772

| Claims | Prior Art |
| --- | --- |
| | provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any |

| Claims | Prior Art |
| --- | --- |
| | digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability |

| Claims | Prior Art |
| --- | --- |
| | to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1314 of 1772

| Claims | Prior Art |
| --- | --- |
| | enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1315 of 1772

| Claims | Prior Art |
| --- | --- |
| | What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses |

| Claims | Prior Art |
| --- | --- |
| | JMS to interact with ICING.<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item; and<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1317 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server; |

| Claims | Prior Art |
| --- | --- |
| | transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 4<br>4. The computer implemented method of claim 1, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1319 of 1772

| Claims | Prior Art |
| --- | --- |
| | from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 19<br>19. The system of claim 16, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>*See also* limitations 1[pre]–1[b], 1[e], 1[g].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| 4. The system of claim 3, wherein the plurality of monitoring criteria Includes a criterion based on magnetic ink character recognition (MICR) line detection. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 3, wherein the plurality of monitoring criteria Includes a criterion based on magnetic ink character recognition (MICR) line detection." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, |

| Claims | Prior Art |
| --- | --- |
| | 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and |

| Claims | Prior Art |
| --- | --- |
| | apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be |

| Claims | Prior Art |
| --- | --- |
| | processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit; |

| Claims | Prior Art |
| --- | --- |
| | provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any |

| Claims | Prior Art |
| --- | --- |
| | digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability |

| Claims | Prior Art |
| --- | --- |
| | to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1327 of 1772

| Claims | Prior Art |
| --- | --- |
| | enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and |

| Claims | Prior Art |
| --- | --- |
| | What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses |

| Claims | Prior Art |
| --- | --- |
| | JMS to interact with ICING.<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item; and<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1331 of 1772

| Claims | Prior Art |
| --- | --- |
| | transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 4<br>4. The computer implemented method of claim 1, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device; |

| Claims | Prior Art |
| --- | --- |
| | from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 19<br>19. The system of claim 16, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>*See also* claim 3.<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| 5. The system of claim 3, wherein the plurality of monitoring criteria Includes criteria based on light brightness, skewing, and warping. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 3, wherein the plurality of monitoring criteria Includes criteria based on light brightness, skewing, and warping." For example:<br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br>*See also* claim 3.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 6. The system of claim 1, wherein the confirming that the mobile check deposit can go forward is performed after the transmitting the digital image of the check from the user's mobile device. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 1, wherein the confirming that the mobile check deposit can go forward is performed after the transmitting the digital image of the check from the user's mobile device." For example:<br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non- |

| Claims | Prior Art |
| --- | --- |
| | Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in |

| Claims | Prior Art |
| --- | --- |
| | software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are |

| Claims | Prior Art |
| --- | --- |
| | called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is |

| Claims | Prior Art |
| --- | --- |
| | authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0167]<br>Deposit bag is a holding/receiving area until the customer commits the deposit. Items are not committed to ICING until the bag is closed;<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item; and<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit |

| Claims | Prior Art |
| --- | --- |
| | checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0185]<br>The CEO system and method then takes the incoming request and access the current session deposit bag 622. It takes the session ID and tags it to each item, having identified the item sequence number. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0186]<br>The CEO system and method checks for any additional validations to be performed 624.<br><br>Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1345 of 1772

| Claims | Prior Art |
| --- | --- |
| | information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 11<br>11. The computer implemented method of claim 1, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 13<br>13. The computer implemented method of claim 2, further comprising the step of:<br>providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 14<br>14. The computer implemented method of claim 1, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in |

| Claims | Prior Art |
| --- | --- |
| | approach is implemented.<br><br>Byrne at claim 15<br>15. The computer implemented method of claim 1, wherein any of said captured one or more objects is replaced by a file upload.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1347 of 1772

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 26<br>The system of claim 16, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 28<br>28. The system of claim 17, further comprising:<br>means for providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 29<br>29. The system of claim 16, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented.<br><br>Byrne at claim 30<br>30. The system of claim 16, wherein any of said captured one or more objects is replaced by a file upload.<br><br>*See also* limitations 1[i], 1[k]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 7. The system of claim 1, wherein the digital image is a modified version of the captured video frame. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 1, wherein the digital image is a modified version of the captured video frame." For example:<br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing |

| Claims | Prior Art |
| --- | --- |
| | image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a |

| Claims | Prior Art |
| --- | --- |
| | method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business |

| Claims | Prior Art |
| --- | --- |
| | goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application;<br>provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method |

| Claims | Prior Art |
| --- | --- |
| | has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as “Okay, the person wrote the check for a hundred dollars. Do you agree? Yes.” Then, instead of keying in 100, the customer can simply click the “yes” button. The scanned images are |

| Claims | Prior Art |
| --- | --- |
| | sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0060]<br>In one embodiment of the invention, the system and method incorporates a detection and notification mechanism. In one embodiment of the invention, the DD Application Server takes certain customer profile information, such as type of scanner, account numbers for which the customer is authorized, etc., and uses the information to detect certain states, such as, if the customer's scanner is plugged in, and appropriately notify the customer. Following is a list of examples of what the detection mechanism can detect: If the scanner is the right version; Was a part of the scanner plugged in the wrong way; if there is a problem with the firmware, i.e. the electronically erasable module on the scanner; etc. The invention can provide a link to the customer from where the customer can download the appropriate helpful information or applications.<br><br>Byrne at [0061]<br>Also, if a new application feature becomes available and, for example, the user has a particular scanner model that has an old version of their basic drivers, the invention can download or direct the customer to a place where the customer can get the driver sets that are designed for the system or application and can download them. The customer can elect to have a system administrator install the feature and appropriate updates or a LAN administrator, and the like. As well, the enterprise can perform the installation, downloads, and upgrades given permission. A clear advantage the invention provides to the enterprise and to the customer is the efficiency provided by the enterprise not having to ship CDs and the like to customers. The installations and upgrades are performed in real-time. Such makes the customer interaction and experience much better.<br><br>Byrne at [0083]–[0084]<br>Interface API<br>Interface API 340 is a Java API library developed for accessing the external systems like WPR 336 and ICING 338. Code is built in the Desktop Deposit application to provide a wrapper to access this |

| Claims | Prior Art |
| --- | --- |
| | functionality. The Interface APIs used by Desktop Deposit are Envoy2 Middleware 330, ICING Middleware 332, IVU [?], and Image Quality Analyzer 334.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0088]<br>This file format is based on the ANSI X9.37 DSTU—2003 "Specifications for Electronic Exchange of Check and Image Data" draft standard with some modifications to accommodate the processing of credits, such as customer Direct Deposit Account (DDA) deposits and adjustments, and to augment sender source identification and file balancing controls.<br><br>Byrne at [0101]–[0103]<br>Interface Between Application and Image Scanner API<br>One embodiment of the invention uses VBScript/JavaScript to communicate with the ActiveX component. The ActiveX component interacts with a customized scanning device API, herein referred to as Ranger API, which in turn interacts with the scanner.<br>For performance consideration a DDScannerServlet 342 which receives the client images, posts the images to the local queue using a business layer. The possibilities of making the queue cluster aware are typically covered during design.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility |

| Claims | Prior Art |
| --- | --- |
| | Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as "default," i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0135]–[0137]<br>ActiveX Component<br>The client side components, herein referred to as CLASP for CEO Lightweight Automated Scanner Process, are developed as ActiveX controls downloadable over the web. The CLASP component provides the interfaces required for the CEO application to interact with a scanner. CLASP provides the abstraction layer required to interact with scanner interface framework, which is Ranger Scanner API.<br>FIG. 4 is a schematic context diagram illustrating how the CLASP component is used in the overall Desktop Deposit solution architecture.<br><br>Byrne at [0138]–[0141]<br>Device abstraction.<br>The Ranger API 402 provides an abstraction layer to the scanner driver 404 allowing applications to interface with different scanner types. |

| Claims | Prior Art |
| --- | --- |
| | Accordingly, the CLASP component 406 uses the Ranger API 402 to talk to the underlying scanner drivers 404 with the understanding that there may be additional interface mechanisms required to be supported by the CLASP component in future.<br>Using the Ranger API 402 alone allows the CLASP component 406 to be lightweight while isolating the CEO application from having to code specifically to the Ranger API.<br><br>Byrne at [0147]<br>The Ranger API 402 is itself packaged as an ActiveX control and hence is used by the CLASP component as an in-process server.<br><br>Byrne at [0151]–[0152]<br>Supported Scanners.<br>In one embodiment of the invention, supported scanners are: Omega System of North America's Panini MyVision X30, X60, and X90 and which support the Silver Bullet Ranger API.<br><br>Byrne at [0153]–[0154]<br>Extensibility.<br>The CLASP component architecture allows for extensibility to support multiple versions of scanner abstraction APIs as well as differences in functionality between these APIs.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said |

| Claims | Prior Art |
| --- | --- |
| | device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 3<br>3. The computer implemented method of claim 1, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 6<br>6. The computer implemented method of claim 2, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 7<br>7. The computer implemented method of claim 1, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 9<br>9. The computer implemented method of claim 1, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for |

| Claims | Prior Art |
| --- | --- |
| | enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 18<br>18. The system of claim 16, wherein said runtime components are any of plugins or ActiveX components.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 21<br>21. The system of claim 17, wherein future plugins are supported that work with a variety of scanning device protocols, scanning devices, and documents.<br><br>Byrne at claim 22<br>22. The system of claim 16, wherein said one or more client components are developed as ActiveX controls that are downloadable over the Web.<br><br>Byrne at claim 23<br>23. The system of claim 16, wherein said one or more client components provide an interface such that said enterprise application interacts with said scanning device. |

| Claims | Prior Art |
|---|---|
| | Byrne at claim 24<br>24. The system of claim 16, wherein said one or more client components provides an abstraction layer to said capturing device such that said enterprise application interfaces with one or more different capturing device types.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>*See also* limitations 1[j], 1[m], 1[o].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 8. The system of claim 1, wherein the wireless network is a cellular network. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 1, wherein the wireless network is a cellular network." For example:<br><br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing |

| Claims | Prior Art |
| --- | --- |
| | image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks |

| Claims | Prior Art |
| --- | --- |
| | using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation;<br>single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, |

| Claims | Prior Art |
| --- | --- |
| | for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1369 of 1772

| Claims | Prior Art |
| --- | --- |
| | amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0185]<br>The CEO system and method then takes the incoming request and access the current session deposit bag 622. It takes the session ID and tags it to each item, having identified the item sequence number.<br><br>Byrne at [0186]<br>The CEO system and method checks for any additional validations to be performed 624.<br><br>Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components; |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1370 of 1772

| Claims | Prior Art |
| --- | --- |
| | from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 11<br>11. The computer implemented method of claim 1, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 12<br>12. The computer implemented method of claim 2, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 13<br>13. The computer implemented method of claim 2, further comprising the step of:<br>providing a deposit bag that serves as a holding area until a deposit is committed. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 14<br>14. The computer implemented method of claim 1, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented.<br><br>Byrne at claim 15<br>15. The computer implemented method of claim 1, wherein any of said captured one or more objects is replaced by a file upload.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1372 of 1772

| Claims | Prior Art |
| --- | --- |
| | detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 26<br>The system of claim 16, wherein said one or more client components interact through HTTP messages.<br><br>Byrne at claim 27<br>27. The system of claim 17, wherein said scanning device returns scanned images and any extracted information to said plugin as Base64 string and plain-text for some selectively extracted regions, and said plugin prepares image and extracted information to the browser as a combination of Base64 strings and plain-text within a HTTP message, which can be further transmitted to said enterprise server component as an HTTP message.<br><br>Byrne at claim 28<br>28. The system of claim 17, further comprising:<br>means for providing a deposit bag that serves as a holding area until a deposit is committed.<br><br>Byrne at claim 29<br>29. The system of claim 16, wherein said capturing one or more objects is performed in parallel with said transmitting said captured one or more objects, whereby a trickle-in approach is implemented.<br><br>Byrne at claim 30<br>30. The system of claim 16, wherein any of said captured one or more objects is replaced by a file upload.<br><br>*See also* limitation 1[j].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| 9[pre]. A system enabling mobile check deposit auto-capture functionality via video frame processing, the system comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] system enabling mobile check deposit auto-capture functionality via video frame processing, the system." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9a] a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the system being configured to perform operations including: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the system being configured to perform operations including." For example:<br>*See* limitations 1[a], 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9b] providing Instructions to the user for capturing an image of a check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing Instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[e]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9d] overlaying an alignment guide on the live video of the check in the field of view of the camera; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9e] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[i]. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1375 of 1772

| Claims | Prior Art |
|---|---|
| providing feedback advising the user that the check has been captured; | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9h] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, the digital image of the check being the captured video frame or being based on the captured video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitation 1[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9i] processing the digital image of the check Including performing optical character recognition and verifying that the digital image of the check Is acceptable for mobile check deposit; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing the digital image of the check Including performing optical character recognition and verifying that the digital image of the check Is acceptable for mobile check deposit." For example:<br>*See* limitation 1[l], 1[n].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9j] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 1[k].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [9k] a bank computer programmed to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a bank computer programmed to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 1[o].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 10[pre]. The system of claim 9, wherein the system is configured to perform additional operations including: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 9, wherein the system is configured to perform additional operations including." For example:<br>*See* limitation 2[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10a] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 2[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [10b] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 2[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 11. The system of claim 9, further comprising one or more computers programmed to perform operations comprising the monitoring, the determining, or both the monitoring and the determining. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 9, further comprising one or more computers programmed to perform operations comprising the monitoring, the determining, or both the monitoring and the determining." For example:<br>*See* limitations 9[a], 9[e], 9[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 12. The system of claim 9, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 9, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* claim 3.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 13. The system of claim 12, wherein the plurality of monitoring criteria Includes criteria based on | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 12, wherein the plurality of monitoring criteria Includes criteria based on light brightness, skewing, and warping." For example:<br>*See* claim 5. |

| Claims | Prior Art |
|---|---|
| light brightness, skewing, and warping. | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 14. The system of claim 9, wherein the confirming that the mobile check deposit can go forward is performed after the transmitting the digital image of the check from the user's mobile device. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 9, wherein the confirming that the mobile check deposit can go forward is performed after the transmitting the digital image of the check from the user's mobile device." For example:<br>*See* claim 6.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 15. The system of claim 14, wherein the user's mobile device receives an Indication of a result of the processing the digital image of the check before the confirming step. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 14, wherein the user's mobile device receives an Indication of a result of the processing the digital image of the check before the confirming step." For example:<br>*See* limitations 1[n], 9[i], 9[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 16. The system of claim 11, wherein the one or more computers include the bank computer. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 11, wherein the one or more computers include the bank computer." For example:<br>*See* limitations 9[a], 9[e], 9f], claim 11.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 17. The system of claim 9, wherein the wireless network is a cellular network. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 9, wherein the wireless network is a cellular network." For example:<br>*See* limitation 9[h]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 18[pre]. A system enabling mobile check deposit auto-capture functionality via video frame processing, the system comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] system enabling mobile check deposit auto-capture functionality via video frame processing, the system." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18a] a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing." For example:<br>*See* limitation 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18b] a software abstraction layer, the software abstraction layer comprising Instructions that, when executed by the processor, control the camera; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a software abstraction layer, the software abstraction layer comprising Instructions that, when executed by the processor, control the camera." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18c] a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera through the | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, the downloaded app comprising Instructions that, when executed by the processor, cause the user's mobile device to perform operations including." For example: |

| Claims | Prior Art |
|---|---|
| software abstraction layer, the downloaded app comprising Instructions that, when executed by the processor, cause the user's mobile device to perform operations including: | *See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18d] providing Instructions to the user for capturing an image of a check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing Instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18e] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18f] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [18g] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18h] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18i] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18j] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 1[k]. |

| Claims | Prior Art |
|---|---|
| (MICR) line on the check is recognized, and an amount payable on the check is validated; and | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18k] a computer comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the computer to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a computer comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the computer to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward." For example:<br>*See* limitation 1[o].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 19[pre]. The system of claim 18, wherein the downloaded app causes the user's mobile device to perform additional operations including: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 18, wherein the downloaded app causes the user's mobile device to perform additional operations including." For example:<br>*See* limitation 2[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [19a] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 2[a]. |

| Claims | Prior Art |
|---|---|
| satisfy the at least one monitoring criterion; and | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [19b] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 2[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 20. The system of claim 18, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 18, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* claim 3.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 21. The system of claim 20, wherein the plurality of monitoring criteria being based on magnetic ink character recognition (MICR) line detection and one or both of corner detection and edge detection. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 20, wherein the plurality of monitoring criteria being based on magnetic ink character recognition (MICR) line detection and one or both of corner detection and edge detection." For example:<br>Byrne at Abstract<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks |

| Claims | Prior Art |
| --- | --- |
| | using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0003]<br>The invention relates to providing a plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment. More particularly, the invention relates to a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account.<br><br>Byrne at [0005]<br>The Check Clearing for the 21st Century Act, better known as Check 21, modernized the U.S. payments system. This new federal legislation, which took effect on Oct. 28, 2004, allows banks to create and present a substitute check or image replacement document (IRD) for payment in place of the original check, accelerating settlement and eliminating many of the costs and risks of physically transporting checks.<br><br>Byrne at [0006]<br>Presently, consumers have available to them the traditional banking visit model. Also, mechanisms for allowing banks to create and present a substitute check exist in the market, however, they require the customer to install a non-browser-based software application, referred to as a fat client. Also, known problems with float, fraud, and check processing speed are present in these mechanisms.<br><br>Byrne at [0007]<br>Currently, there are systems and methods that relate to check deposit and image processing at locations other than a bank, as follows. M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Pat. No. 6,450,403 B1 (Sep. 17, 2002) and M. Martens, C. P. Tresser, R. J. Von Gutfeld, and C. W. Wu, Method and Apparatus for Depositing Ordinary Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0084321 A1 (Jul. 4, 2002) discuss an apparatus and method that allow depositing ordinary checks from home or office. A special scanner is used to scan an endorsed check for deposit. The check may have printed thereon encryptions in at least selected locations. Scanning the endorsed check with the scanner generates a digitized version of the check. The scanner virtually partitions the digitized version of the check into a plurality of regions. These regions may be stripes or zones. Each region is successively |

| Claims | Prior Art |
| --- | --- |
| | examined to extract from the digitized version of the check information from that region. The information extracted from a region is encrypted and transmitted to a bank. Upon acknowledgment from the bank, at least some of the regions of the plurality of regions are voided by a form of indelible but non-invasive, e.g. allowing reading after voiding, marking, such as punched holes, burned areas, overprinting of a pattern. The processing by the scanner continues until all regions have been processed. This progressively voiding scheme is an important security feature. The scanner is preferably contained in a secure housing with an encryption processor and the housing preferably includes an arrangement for detecting tampering and preventing normal operation if tampering is detected.<br><br>Byrne at [0008]<br>J. E. Jones, P. A. Jones, R. M. Gafron, W. J. Jones, and D. U. Mennie, Image Processing Network, U.S. Patent Application Publication No. US 2002/0001393 A1 (Jan. 3, 2002) discuss a paperless image processing network where images and other information are exchanged between various financial institutions rather than physical documents. Further discussed is an image processing network where images and other information as well as physical documents are exchanged between various financial institutions. Also discussed is an image processing network where the processing of document images is distributed amongst various locations rather than concentrated at a single central location. Also discussed is an image processing network which is capable of processing both images and physical documents; an image processing network which is capable of extracting the MICR, CAR, and LAR data from checks and inserting this data into an image file or files; an image processing network where images are sorted between transit and on-us image files; and a data file format that comprises two subparts, containing image data and the other part containing data extracted from the document's image.<br><br>Byrne at [0009]<br>M. Martens and C. P. Tresser, Method and Apparatus for Depositing Paper Checks from Home or Office, U.S. Patent Application Publication No. US 2002/0065786 (May 30, 2002) discuss a method and apparatus in combination with a special form of paper checks, which allows for the secure deposit of paper checks from home or office; in other words, at a location other than the bank or an ATM. The apparatus can be implemented at the payee's home or office with a Personal Computer (PC) which has a scanner attached to it and connected to the World Wide Web (WWW) on the Internet. The process of depositing paper checks begins by the payee endorsing a check having printed thereon encryptions in at least selected locations where information is written by a payer, the act of writing by the payee obscuring some of the encryptions. The payee then scans the endorsed check with a scanner to generate a digitized |

| Claims | Prior Art |
| --- | --- |
| | version of the check. The computer extracts from the digitized version of the check a concatenated branch number, account number and check number and a corresponding digital signature. The payee then transmits the extracted information together with the digitized version of the check for deposit. The checks are specially designed to prevent fraud such as alterations of the payee, amount and multiple deposits. In addition to the encryptions imprinted on the check, a secret key and a plurality of digital signatures are generated based on the concatenated branch number, account number and check number.<br><br>Byrne at [0010]<br>Ravi V. Acharya and J. Kang, System and Method for Electronic Deposit of Third-Party Checks by Non-Commercial Banking Customers from Remote Locations, International Publication No. WO 01/61436 A2 (23 Aug. 2001) discuss a system and method for initiating a deposit transaction, where the depositor is a non-commercial banking customer located at a remote location, and where the item to be deposited is a paper check from a third party, payable to the depositor. Features include a Remote Customer Terminal (RCT) with certain input devices, connected to a bank system. The image and/or other data sent from the RCT to the Bank of First Deposit (BOFD) may be processed by conversion to Electronic Funds Transfer (EFT), via Electronic Check Presentment (ECP), or via check reconstruction.<br><br>Byrne at [0011]<br>A. Slater, M. L. Sears, H. P. Rin-Rin, D. C. DO, H. P. McSharry, E. M. R. Dudasik. S. M. Gryte, and R. O. Brooks, System and Method for Image Depositing, Image Presentment and Deposit Taking in a Commercial Environment, European Patent Application No. EP 0 984 410 A1 (08 Mar. 2000) discuss a method and system for scanning a check and/or cash to create an electronic image of the front and the back of the check and/or cash. The image is then processed and transmitted electronically to a central location. The image may be recreated into a paper form at the central location, resembling the original paper check or cash. Paperless processing of checks and cash is thus provided, including local voiding and storage of the check without requiring immediate pickup, while still allowing the transaction to be processed.<br><br>Byrne at [0012]<br>However, none of the references cited teach or suggest creating efficiencies by providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account, while, at the same time, provide a light client vs. heavy client side footprint/solution and provide means for a customer to install software via application-managed |

| Claims | Prior Art |
| --- | --- |
| | download.<br><br>Byrne at [0013]<br>It would be advantageous to provide a system and method for capturing data objects from a programmed machine where the configuration is not limited to a fixed location, office, or home personal computer; where a supported capturing device, such as a scanner, is plugged into the programmed machine, e.g. a personal computer, at any location that works with downloadable runtime components, such as plug-in software, and can be used to capture data objects, e.g. to scan checks and prepare deposits.<br><br>Byrne at [0016]<br>It further would be advantageous to provide a system and method that displays substitute checks or copies of substitute checks online when a customer requests check copies or view images online.<br><br>Byrne at [0017]<br>It further would be advantageous to provide a system and method that allows a bank customer to be able to capture electronic images of the customer's checks and to transmit the checks for processing instead of requiring the customer to physically transport the checks to the bank for deposit.<br><br>Byrne at [0018]–[0032]<br>It further would be advantageous to provide a system and method that attains the following business goals:<br>provide faster access to funds and information;<br>reduce transportation time and costs;<br>leverage Check 21 legislation;<br>provide alternative to traditional banking visit;<br>provide low cost entry point to market model; develop, deploy, and manage such; and<br>enrich customer experience.<br>It further would be advantageous to provide a system and method that attains the following technical goals:<br>provide a light client vs. heavy client side footprint/solution;<br>provide means for a customer to install software via application-managed download;<br>provide means for a customer to scan checks and upload them to the application; |

| Claims | Prior Art |
| --- | --- |
| | provide an application that validates checks and creates deposit transactions;<br>provide an application that sends deposit transactions to an existing downstream item processing;<br>provide for the customer access to timely information on the status of the customer's deposit or any errors which would prevent an item from being accepted; and<br>provide a same and/or better than response time.<br><br>Byrne at [0033]<br>A browser plug-in to drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0041]<br>A browser software plug-in to provide an interface between the user and an enterprise and drive a device for securely transmitting documents in real-time over the Internet into an enterprise's computing environment is provided. Leveraging the Check Clearing for the 21st Century Act, a system and method for providing a browser plug-in that allows a bank customer to scan checks using a personal computer and deposit the checks via the browser to a bank account is provided.<br><br>Byrne at [0042]<br>It should be appreciated that for the purposes of understanding the invention, the discussion herein involves a personal computer, a scanner, scanned image, a deposit post-process, and the like. However, such specific references herein are by way of example only and are not meant to be limiting. The spirit and scope of the invention covers any thin client with respect to any enterprise post processing and any type of capturing device and captured data object, where here an object includes, but is not limited to, any digitized image, portion of an image, data stream, screen, or window.<br><br>Byrne at [0043]<br>In one embodiment of the invention, five major components are provided to work in a manner that allows the user to scan, transmit, and deposit checks or documents to the receiving organization electronically. A personal computer type computer (PC), application software provided by the enterprise or third-party supplier (application software), a scanner supported by the application software (scanner), scanner drivers |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1389 of 1772

| Claims | Prior Art |
| --- | --- |
| | provided by the scanner manufacturer or third-party (device drivers), and a receiving software application to process the incoming images and data (enterprise application).<br><br>Byrne at [0048]<br>The CEO Portal, C21 app as well as the plug-in are resident on the enterprise servers. The user launches a session for scanning checks from the browser window running the plug-in(s). Scanner interface APIs are called from the browser plug-in(s) for communicating with a scanning device that is communicably connected to the PC 106. The scanner APIs make device driver calls to the scanner, which in this example is RDM 6000. The user proceeds with scanning one or more checks through the device. Each check can be marked with indelible ink, if the scanner is so equipped, indicating that the check has been electronically processed. Such scanned images are transmitted by the browser window running the plug-in(s) to a check image repository communicably connected to the server desktop deposit application 110. The desktop deposit application, C21 App is used to manage the checks which are launched. Through a viewer and editing application, which in this example is referred to as C21 Viewer/Editor Portlet, which is a part of C21 App, a user can view and edit commands related to images stored on the check image repository that had been previously scanned 112. Likewise, through C21 Viewer/Editor Portlet, a user can perform various functions, such as fetch, edit, and delete on the scanned images 114.<br><br>Byrne at [0049]<br>The invention provides flexibility as follows. In one embodiment of the invention, the system and method has the capability to perform version checks and updates. In one embodiment of the invention, the system and method is modular; it employs framework adaptability to support future plug-ins as necessary to work with a wide variety of proprietary check-reader device protocols. In one embodiment of the invention, the system and method provides the capability of the self-detection of installed scanner device and capability to guide the user of related events and detects necessary downloads, upgrades, and patches.<br><br>Byrne at [0050]–[0056]<br>One embodiment of the invention provides a helper application that facilitates the following:<br>scanning of checks, front and back;<br>MICR decoding;<br>compression and encryption;<br>allowing user to enter check amount and annotation; |

| Claims | Prior Art |
| --- | --- |
| | single submission and batching of scanned checks; and<br>uploading one or a batch of checks to a staging repository via database and/or FTP uploads.<br><br>Byrne at [0058]<br>An interface architecture system and method according to one embodiment of the invention can be described with reference to FIG. 2. A desktop deposit customer, shown here as a CEO customer 202, is authenticated and authorized by applications on the desktop deposit server, shown here as DesktopDeposit (DD) Application Server 204. A second server 206 is shown between two firewalls and serves to provide a secure interface between external applications and those internal to an enterprise, such as Wells Fargo, for example. The customer proceeds to scan one or more checks through the check scanner device 208. The customer uploads the scanned check images to the DD application server for post processing. The uploaded images are stored, could be temporarily, in a database, shown in FIG. 2 as Check Image Work in Progress (WIP) 210 for review and disposition. The images are also sent, for example, via an wfXML service, to an Image Inspector Services 211, referred to herein as Automated Image Metrics (AIM). AIM is a placeholder for certain basic services. For example, AIM can perform character recognition on a image, e.g. courtesy amount recognition and legal amount recognition (CARLAR) processing. For instance, a customer might write an amount of X in the courtesy box and an amount of Y in the legal amount box resulting in a mismatch. The system and method can respond by sending such information back to the customer, informing the customer of such mismatch. As well, the system and method can provides hints, such as "Okay, the person wrote the check for a hundred dollars. Do you agree? Yes." Then, instead of keying in 100, the customer can simply click the "yes" button. The scanned images are sent to a second database, shown here as Inbound Check Image Gateway (ICING) 212, to be checked again for quality and customer disposition. The DD Application Server 204 requests or sends a message query to ICING 212 for file validation and response, and possibly in batch mode. DD Application Server sends user profile information, including account information, product profile information, and the like, to enterprise backend servers. One such server is shown in the figure as Envoy 214. Customer billing information is sent to a backend system, shown here as Tandem Billing 216.<br><br>Byrne at [0085]–[0087]<br>Interface Between Application and ICING<br>Desktop Deposit Application Interfaces with ICING Using MQ Series.<br>The images of checks transmitted with the file are retained and archived by the enterprise, i.e. in this example, Wells Fargo bank (WFB), for use in research and adjustments and for legal compliance |

| Claims | Prior Art |
| --- | --- |
| | purposes. These images are also made available for customer retrieval, through associated WFB information reporting products. Each incoming item is assigned a unique item sequence number (ISN), based on Wells Fargo standards and including a parameter-driven sender identifier. The ISN is included with the related item record in file and should be printed, if possible, on the paper item by the check transport device to support easy retrieval of the original item from temporary storage when necessary.<br><br>Byrne at [0104]–[0119]<br>Image Viewer Utility<br>Image viewer utility helps the user in viewing the image in an appropriate image size. It provides following functionality to the user:<br>Ability to lighten an image, default increment should be +25;<br>Ability to darken the image, default increment should be −25;<br>Ability to invert an image;<br>Ability to rotate the image in increments of 90 degrees in both directions;<br>Ability to zoom in or out on the image; and<br>Ability to scroll left, right, up or down on an image any time the full image is not visible in the window.<br>Interface Between Application and Image Viewer Utility<br>The invoking application is able to determine and adjust the following:<br>How many image viewer windows to display on the page, typically one or two;<br>What operations to apply to the image before displaying as “default,” i.e. shrinking, lightening, rotating, etc.;<br>What the maximum zoom level should be, such as in terms of a percent of the original image size;<br>What the zoom increment is;<br>What the lighten/darken increment is; and<br>What options to display besides Image Print and Download, for example on a bar below the image viewing window and such as showing or hiding the View Full Size link as desired.<br><br>Byrne at [0155]–[0156]<br>An Exemplary Scanning Device<br>One embodiment of the invention provides means for building a device wrapper or Ranger-wrapper ActiveX that can return images as Base64 strings. It provides a control that can take scanned image(s) and package them up into a string that can be transmitted to the enterprise server, such as CEO. One |

| Claims | Prior Art |
| --- | --- |
| | embodiment of the invention provides a means for transmitting them to CEO that includes setting up the images using a Base64 encoding scheme and transmitting the images to CEO as a standard HTML form-post. The invention provides a means for the control or device that takes the current image as an argument, such as front-1, front-2, reverse-1, or reverse-2, and returns a Base64 encoded string. In one embodiment of the invention, the code returns just the Base64 string without any MIME information. It is possible to prefix the string with additional information as well as return just the raw Base64 string.<br><br>Byrne at [0161]<br>In step 1.1, a user clicks on a button on a browser screen in Desktop Deposit. The javascript submits the form data to the Desktop Deposit gateway servlet, which invokes Action Servlet, along with the Action. In step 1.2, the Action Servlet invokes the Request Processor class. In step 1.3, the Request Processor invokes the appropriate Action Class and sets Action Form. In step 1.4, the Action Class builds the appropriate domain object from the data. It does a look-up, calls an appropriate Business Object (BO), invokes the appropriate method on it, and passes the domain object to it. In step 1.5, the BO evaluates the business rules, if any, calls one or more Data-Access Object, invokes appropriate method on it, and passes the domain object(s) to it. In step 1.6, the DAO makes one or more Interface API method calls to the access to the Database, Envoy2. In step 1.7, tor Database Calls, JDBC is the communication mechanism for accessing Local Desktop Deposit database. In step 1.8, for WPR access, the Envoy2 Client talks to Envoy2 Server hosted externally with appropriate protocol. In step 1.9, the JSP extracts data from the Action Form and builds the screen HTML and sends it to the browser. In step 2.1, VBScript/JavaScript is used for interacting with ActiveX, this will in turn iterface with Scanner. In step 2.2, Actions or steps 1.2, 1.3, 1.4, 1.5, 1.6 are performed. In step 2.3, Business Layer interacts with JMS layer to put the data in JMS queue. In step 2.3, Image Viewer Utility is used for manupulation of the checks. In step 2.4, Image Quality Analyser uses wfxml protocol to interact with IQA Service. And, in step 2.5, Interface API uses JMS to interact with ICING.<br><br>Byrne at [0170]<br>The client can elect to have the OCR information and/or image sent back from the server as an acknowledgment for each scanned item; and<br><br>Byrne at [0173]<br>Referring to FIG. 6, a schematic flow diagram, an exemplary client flow according to the invention can be |

| Claims | Prior Art |
| --- | --- |
| | described. A user begins by logging into a CEO system. A Check21 application launches in response and creates an account list from the user's authorized accounts. The user selects an account to which to deposit checks to, e.g. from a dropdown list. The system creates a bag for this session 602.<br><br>Byrne at [0174]<br>An application child or window opens and is an ActiveX control. The logic following (from 606 through 614) can be performed by a full application or can be part of an ActiveX control. Regardless, the front of an image is scanned 606. The check amount is obtained, the MICR is OCRed, and it is verified that the MICR is magnetic.<br><br>Byrne at [0175]<br>The image back is scanned 608. Optionally, it is detected whether the check is endorsed. The system uses a checking means, such as checking for an imprint image, a scanned serial number, and the like, to indicate that the check had been properly scanned.<br><br>Byrne at [0176]<br>An HTTP POST is prepared 610. Amount is formatted, MICR information, such as RTN, account, amount, RAA, and the like, is formatted. This may be a unique hash.<br><br>Byrne at [0177]<br>Image front and image back are added to a POST message. It is optional to take in customer free-form text information 612.<br><br>Byrne at [0183]<br>The CEO Check21 Servlet filters input to check if the message is complete 618. It rejects partial POSTs immediately. The total turn-around time is preferably one second or better per check item.<br><br>Byrne at [0189]<br>An acknowledgment/post receipt of each item is sent, such as echo RAA 630. Optionally, an echo front, back image, and/or receipt information is sent. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at [0191]<br>The client flow after the client elects to commit uploaded contents can be described with reference to FIG. 7, a flow diagram showing the same. Via an application child window it portlet, a user selects a finish scanning means. The finish function closes the bag and commits to a deposit. The message is sent to the Check21 servlet 704. The servet adds a deposit ticket means as a trailer to the bag 706. A message is posted to ICING to pick up the bag and process a deposit batch 708. The servlet responds back to the user with the deposit ticket information 710.<br><br>Byrne at claim 1<br>1. A computer implemented method for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising the steps of:<br>one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, said enterprise server launching an enterprise application and downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>capturing one or more objects through said capturing device;<br>from said one or more runtime components, optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and passing said detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components. |

| Claims | Prior Art |
| --- | --- |
| | Byrne at claim 4<br>4. The computer implemented method of claim 1, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 5<br>5. The computer implemented method of claim 2, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>Byrne at claim 16<br>16. A system for transmitting captured objects from a capturing device in real-time from one or more client components through an enterprise server component to one or more enterprise backend processing components, comprising:<br>means for one or more client components creating a session with a component on an enterprise server;<br>responsive to said creating a session, means for said enterprise server launching an enterprise application and means for downloading one or more corresponding runtime components onto said one or more client components;<br>from a browser window on said one or more client components, means for executing said one or more runtime components to invoke interface APIs of a capturing device, and on said browser window means for displaying an interface to said enterprise application for accessing said enterprise application;<br>from the browser runtime components, means for calling capturing device interface APIs to communicate with said device, wherein said capturing device interface APIs make device driver calls to said capturing device;<br>means for capturing one or more objects through said capturing device;<br>from said one or more runtime components, means for optionally calling capturing device interface APIs to direct said device to selectively extract additional information from each said captured object;<br>means for said browser runtime components pre-processing said captured objects to ensure message consistency and compatibility with said enterprise server;<br>means for transmitting said captured one or more objects and, if available, said selectively extracted additional information, to a repository communicably connected to said enterprise application for enterprise backend post-processing; and<br>means for said browser runtime components detecting exception conditions on said capturing device and, if within scope of said runtime components, handling exception conditions, and means for passing said |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1396 of 1772

| Claims | Prior Art |
|---|---|
| | detected exception conditions and said handled exception conditions, as well as unhandled exception conditions and ancillary information to said enterprise server component;<br>wherein no captured objects are stored on said one or more client components.<br><br>Byrne at claim 17<br>17. The system of claim 16, wherein said capturing device is a scanning device and said one or more captured objects is one or more scanned images.<br><br>Byrne at claim 19<br>19. The system of claim 16, wherein said selectively extracted information is any of: a Magnetic Ink Character Recognition (MICR) line, written text, or a bar-code.<br><br>Byrne at claim 20<br>20. The system of claim 17, wherein any of said scanned one or more images for deposit is a bank check, coupon, or document.<br><br>*See* claims 3, 4, 20.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 22. The system of claim 20, wherein the plurality of monitoring criteria Includes criteria based on light brightness, skewing, and warping. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 20, wherein the plurality of monitoring criteria Includes criteria based on light brightness, skewing, and warping." For example:<br>*See* claim 5.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| 23. The system of claim 18, wherein the confirming that the mobile check deposit can go forward is performed after the transmitting the digital image of the check from the user's mobile device. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 18, wherein the confirming that the mobile check deposit can go forward is performed after the transmitting the digital image of the check from the user's mobile device." For example:<br>*See* claim 6.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 24. The system of claim 18, wherein the digital image is a modified version of the captured video frame. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 18, wherein the digital image is a modified version of the captured video frame." For example:<br>*See* claim 7.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 25. The system of claim 18, wherein the wireless network is a cellular network. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 18, wherein the wireless network is a cellular network." For example:<br>*See* claim 8.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 26[pre]. A system enabling mobile check deposit auto-capture functionality via video frame processing, the system comprising: | To the extent the preamble is limiting, Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] system enabling mobile check deposit auto-capture functionality via video frame processing, the system." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [26a] a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing." For example:<br>*See* limitation 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26b] a software abstraction layer, the software abstraction layer comprising Instructions that, when executed by the processor, control the camera; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a software abstraction layer, the software abstraction layer comprising Instructions that, when executed by the processor, control the camera." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26c] a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, the downloaded app comprising Instructions that, when executed by the processor, cause the user's mobile device to perform operations including: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, the downloaded app comprising Instructions that, when executed by the processor, cause the user's mobile device to perform operations including." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26d] providing Instructions to the user for capturing a check; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing Instructions to the user for capturing a check." For example:<br>*See* limitation 1[d]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26e] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26f] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26g] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26h] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example: |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1400 of 1772

| Claims | Prior Art |
|---|---|
| one monitoring criterion, providing feedback advising the user that the check has been captured; | *See* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26i] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26j] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 18[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26k] the system configured to perform image processing operations on an image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "the system configured to perform image processing operations on an image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device." For example:<br>*See* limitations 1[j], 1[n]. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1401 of 1772

| Claims | Prior Art |
|---|---|
| of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device; and | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26l] a computer programmed to update a balance of an account at a financial Institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a computer programmed to update a balance of an account at a financial Institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 1[o].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 27[pre]. The system of claim 26, wherein the downloaded app causes the user's mobile device to perform additional operations including: | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 26, wherein the downloaded app causes the user's mobile device to perform additional operations including." For example:<br>*See* limitation 2[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [27a] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 2[a]. |

| Claims | Prior Art |
|---|---|
| live video of the check does not satisfy the at least one monitoring criterion; and | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [27b] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 2[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 28. The system of claim 26, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 26, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* claim 3.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 29. The system of claim 28, wherein the plurality of monitoring criteria Includes a criterion based on magnetic ink character recognition (MICR) line detection. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 28, wherein the plurality of monitoring criteria Includes a criterion based on magnetic ink character recognition (MICR) line detection." For example:<br>*See* claim 4.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
| --- | --- |
| 30. The system of claim 26, wherein the wireless network is a cellular network. | Byrne discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 26, wherein the wireless network is a cellular network." For example:<br><br>*See* claim 8.<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Byrne alone; (2) Byrne in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1404 of 1772

**Exhibit B18 - Invalidity of 12,211,095 in view of Ranjan**

Using a Camera with Windows Mobile 5 by Amit Ranjan ("Ranjan") anticipates and/or renders obvious (alone or in combination with other references), at least under USAA's apparent infringement theories, all asserted claims of U.S. Patent No. 12,211,095 ("the '095 patent"). Ranjan was published or otherwise publicly available by July 21, 2006 and is therefore prior art to the '095 patent.

Regions has charted the portions of the reference that correspond to each element, including at least under USAA's apparent interpretation of the claims as reflected in USAA's Complaint and Infringement Contentions. Where the chart below states that the reference "discloses," "teaches," or "suggests" a claim limitation, such disclosure, teaching, or suggestion may be express, implied, or inherent. The cited evidence is merely illustrative, and Regions reserves the right to cite alternative or additional evidence.

No part of this exemplary chart construes, or is intended to construe, the specification, file history, or claims of the asserted patents. In addition, nothing stated in this chart shall be treated as an admission or suggestion that Regions agrees with USAA regarding either the scope of any of the asserted claims or Regions' apparent interpretation of the scope of the asserted claims, or that Regions' accused products meet any limitations of the claims.

To the extent that USAA contends that Ranjan does not disclose one or more limitations of the asserted claims, it would have been obvious to combine the teachings of Ranjan with (1) the knowledge of one of ordinary skill in the art, and/or (2) the teachings of any of the prior art set forth in Regions' other invalidity charts and cover pleading, to show all the limitations of the asserted claims, including for the reasons set forth in Regions' Invalidity Contentions Cover Pleading, which are incorporated herein.

The cited disclosure further shows that it would have been obvious to combine Ranjan with the prior art set forth in Regions' other invalidity charts and cover pleading, for example by showing similarities with the other prior art and/or elements that could be improved by combination, replacement, and/or modification in view of the other prior art.

Because USAA has yet to identify any limitation of the asserted claims that it contends is not fully disclosed by Ranjan, either alone or in combination with other prior art cited by Regions, Regions expressly reserves the right to rebut any such contention, including by identifying additional § 103 combinations, if any such contention is made by USAA.

| **Claims** | **Prior Art** |
|---|---|
| 1[pre]. A system enabling mobile check deposit auto-capture functionality via video frame processing, the system comprising: | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] system enabling mobile check deposit auto-capture functionality via video frame processing, the system." For example: |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1405 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br><br>Ranjan at Page 1<br>  <br> |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1407 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1408 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1410 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1413 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

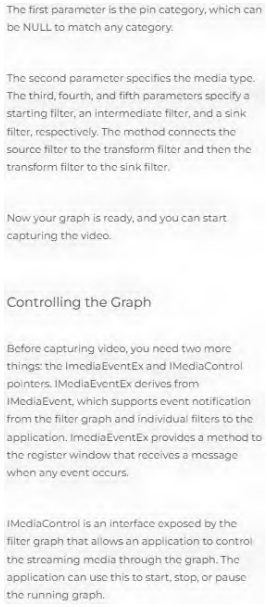
&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |



The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1416 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

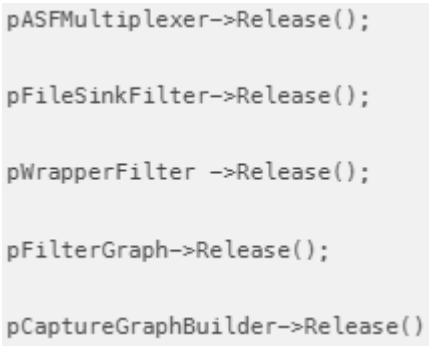
    &dwStart, &dwEnd, wStartCookie, wEndC
```

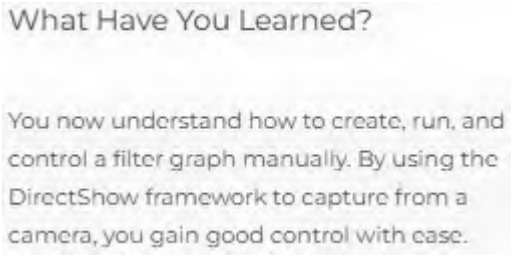
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.

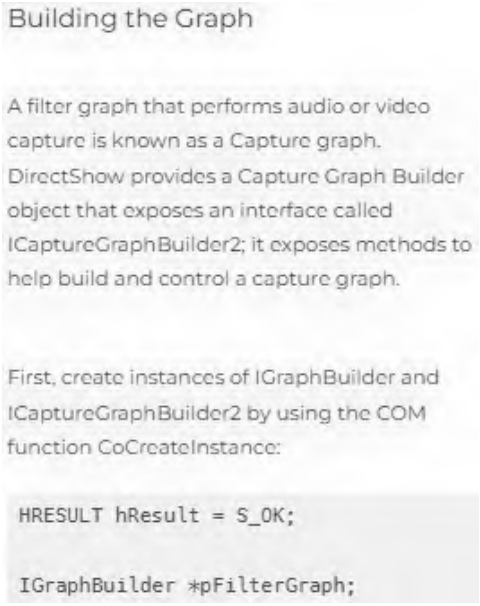

USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1419 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1a] a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing." For example:<br>Ranjan at page 1<br>  <br> |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at Page 1<br>  <br><br>Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |



Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1423 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

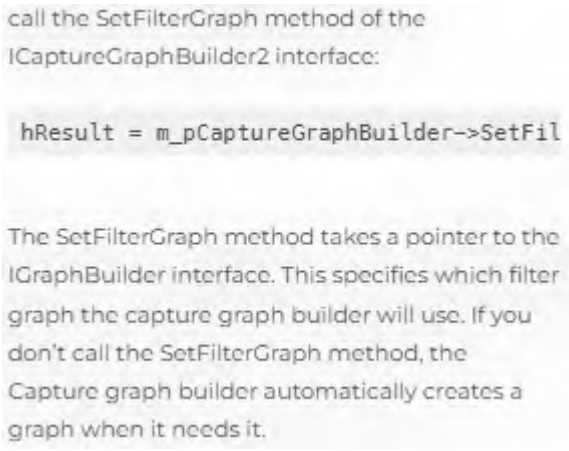
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |

The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

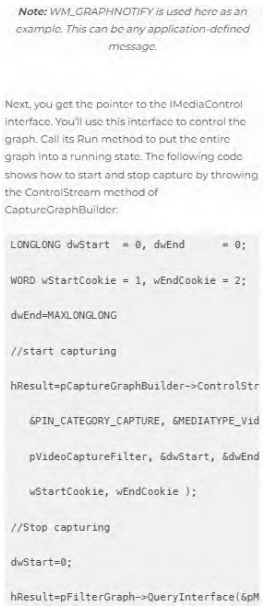
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

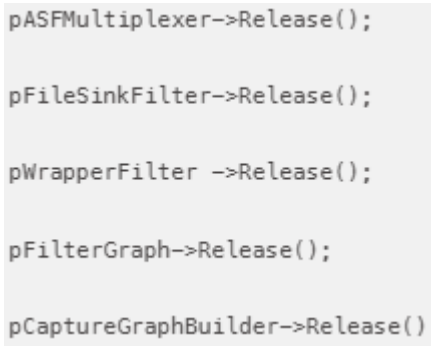
    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

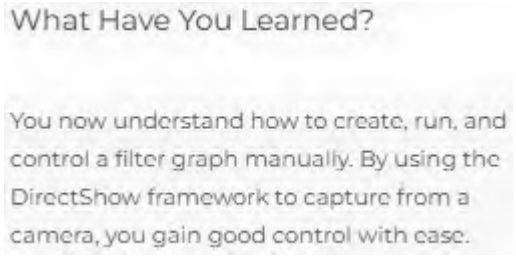
    &dwStart, &dwEnd, wStartCookie, wEndC
```

The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1b] a software abstraction layer for controlling the camera, the software abstraction layer comprising Instructions that, when executed by the processor, control the camera; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a software abstraction layer for controlling the camera, the software abstraction layer comprising Instructions that, when executed by the processor, control the camera." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1438 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1439 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1444 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

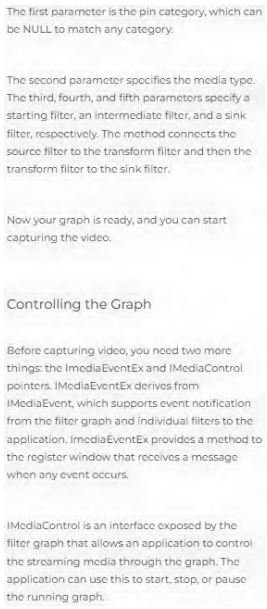
&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1446 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10  The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

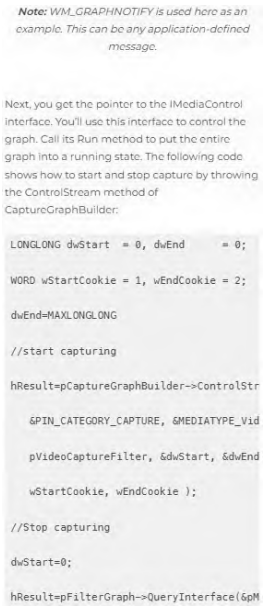
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

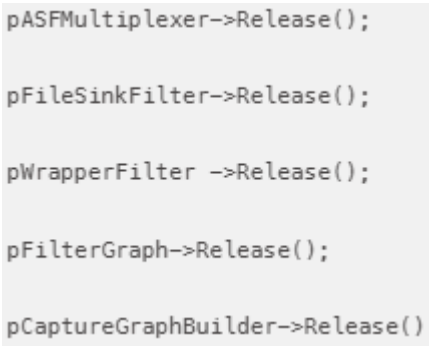
    &dwStart, &dwEnd, wStartCookie, wEndC
```

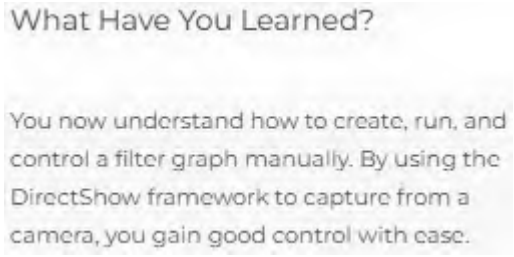
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1450 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1c] a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, the downloaded app comprising Instructions that, when executed by the processor, cause the user's mobile device to perform operations including: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, the downloaded app comprising Instructions that, when executed by the processor, cause the user's mobile device to perform operations including." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1454 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1455 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1456 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1457 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

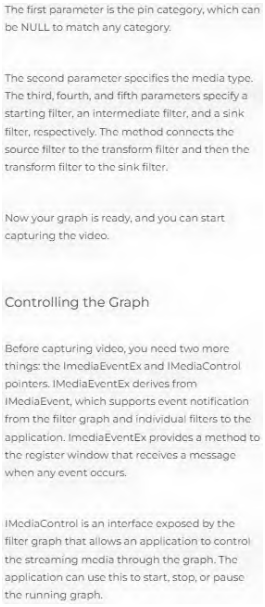
```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10  The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

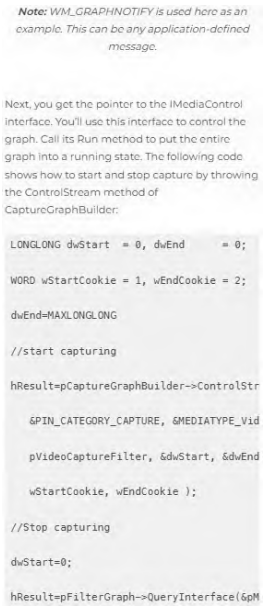
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

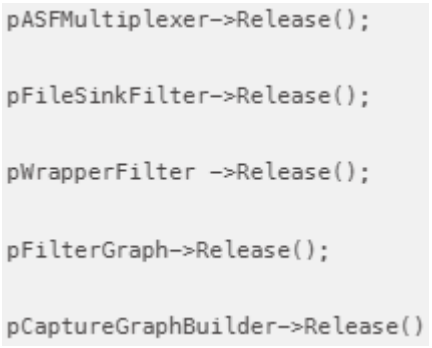
    &dwStart, &dwEnd, wStartCookie, wEndC
```

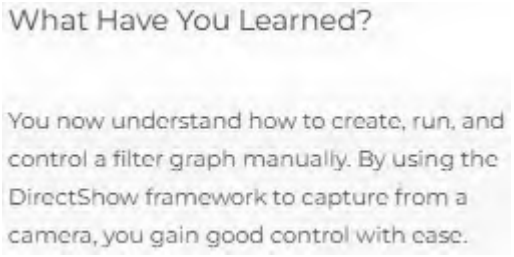
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1466 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre]–1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1d] providing Instructions to the user for capturing an image of a check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing Instructions to the user for capturing an image of a check." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Camera Driver (DMO)<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1471 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1476 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |

The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1479 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

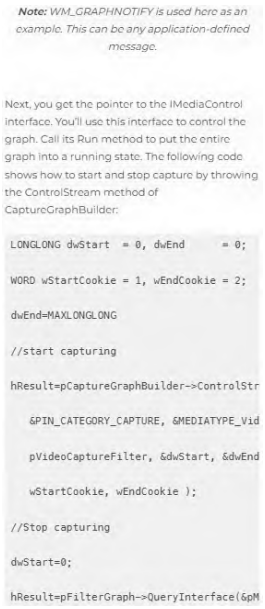
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

    &dwStart, &dwEnd, wStartCookie, wEndC
```

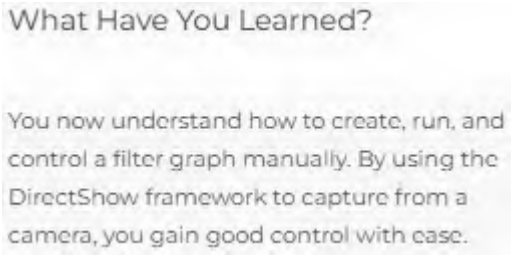
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1482 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1e] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1487 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

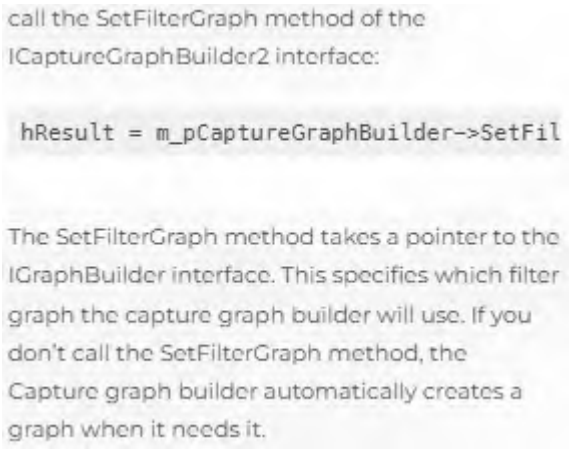
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>DEVMGR_DEVICE_INFORMATION devInfo;<br>CComVariant CamName;<br>CPropertyBag PropBag;<br>GUID guidCamera = { 0xCB998A05, 0x122C,<br>0x3E, 0x4D, 0x7E, 0x<br>devInfo.dwSize = sizeof(devInfo);<br>FindFirstDevice( DeviceSearchByGuid, &gu<br>CamName=devInfo.szLegacyName<br>PropBag.Write( _T("VCapName"), &CamName<br>pPropertyBag->Load( &PropBag, NULL );<br>hResult =pFilterGraph->AddFilter( pVideo<br>_T("Vi<br>Ranjan at page 7<br>pPropertyBag.Release() |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1492 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

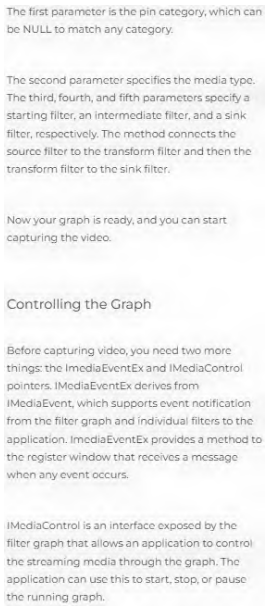
&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1494 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10  The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

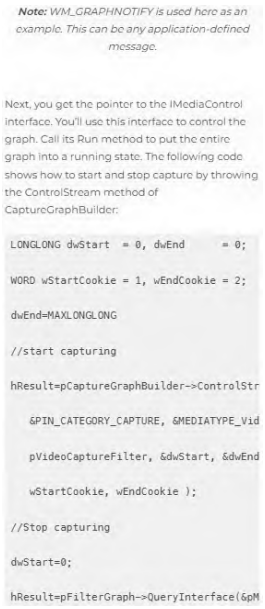
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

    &dwStart, &dwEnd, wStartCookie, wEndC
```

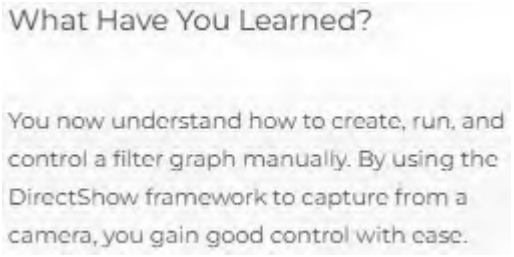
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1498 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre]–1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1f] overlaying an alignment guide on the live video of the check in the field of view of the camera; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Camera Driver (DMO)<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1503 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1507 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1508 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1510 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |

The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

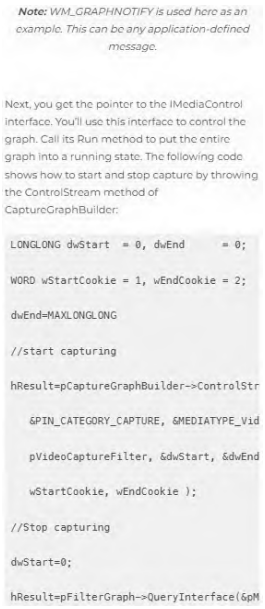
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

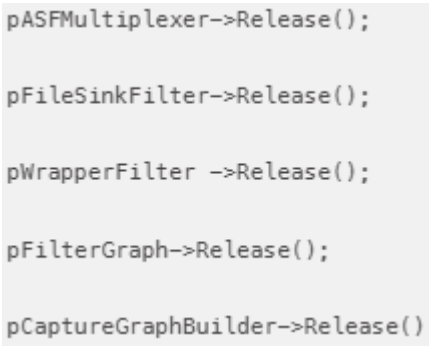
    &dwStart, &dwEnd, wStartCookie, wEndC
```

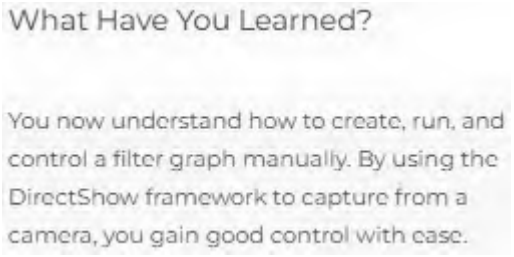
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1514 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre]–1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1g] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br><br>Ranjan at Page 1<br>  <br> |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1519 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1523 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1524 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

&pFileSinkFilter );
```

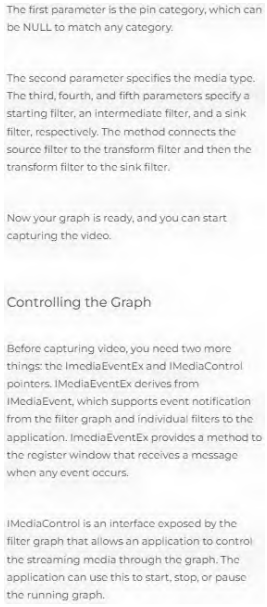
You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1526 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |



The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1527 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

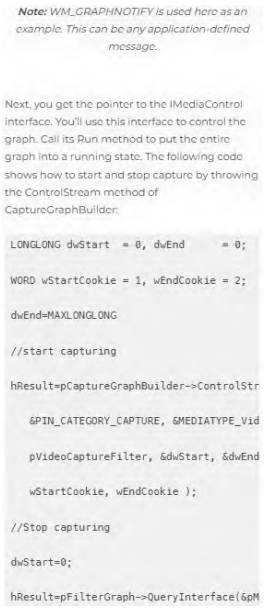
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

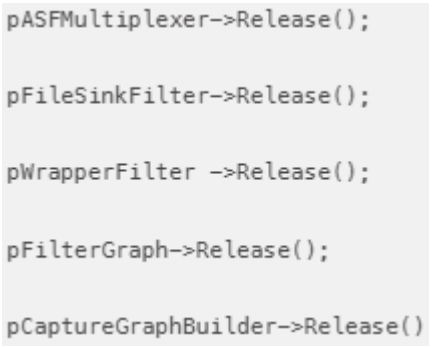
    &dwStart, &dwEnd, wStartCookie, wEndC
```

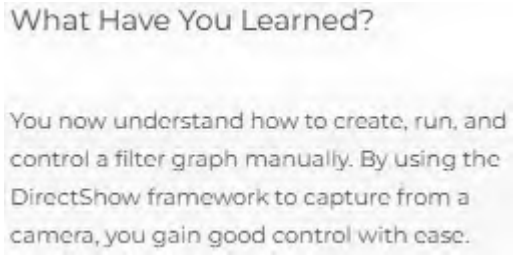
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.

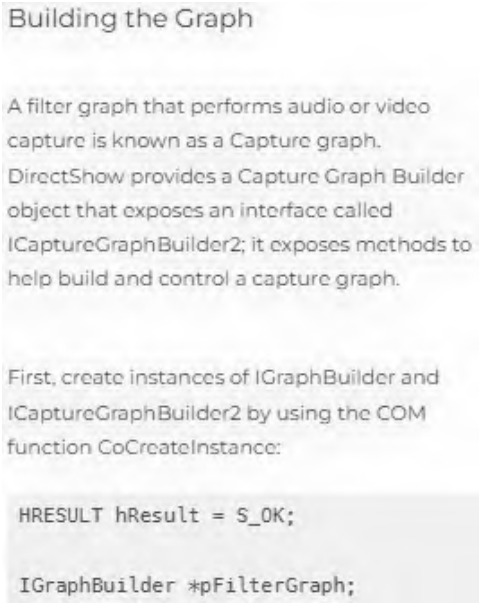

USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1530 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre]–1[b], 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1h] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3<br>  <br>Building the Graph<br>A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.<br>First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:<br>`HRESULT hResult = S_OK;`<br>`IGraphBuilder *pFilterGraph;` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

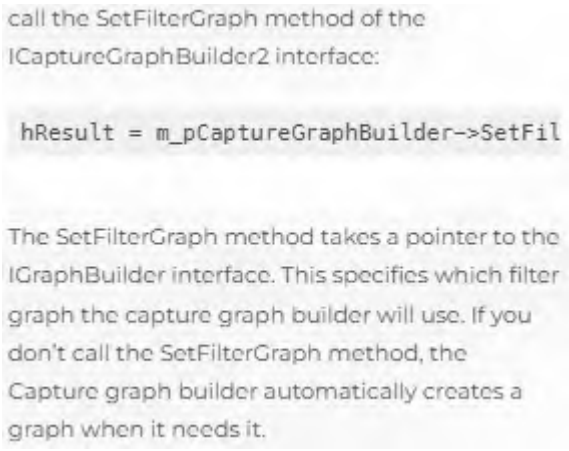
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>DEVMGR_DEVICE_INFORMATION devInfo;<br>CComVariant CamName;<br>CPropertyBag PropBag;<br>GUID guidCamera = { 0xCB998A05, 0x122C,<br>0x3E, 0x4D, 0x7E, 0x<br>devInfo.dwSize = sizeof(devInfo);<br>FindFirstDevice( DeviceSearchByGuid, &gu<br>CamName=devInfo.szLegacyName<br>PropBag.Write( _T("VCapName"), &CamName<br>pPropertyBag->Load( &PropBag, NULL );<br>hResult =pFilterGraph->AddFilter( pVideo<br>_T("Vi<br>Ranjan at page 7<br>pPropertyBag.Release() |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1540 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

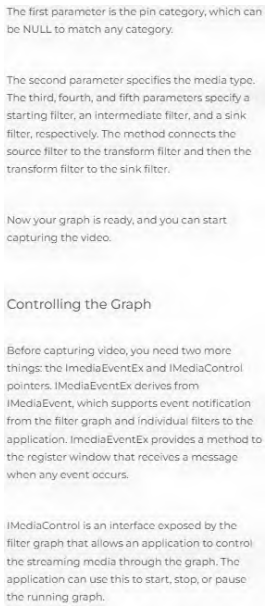
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10  The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

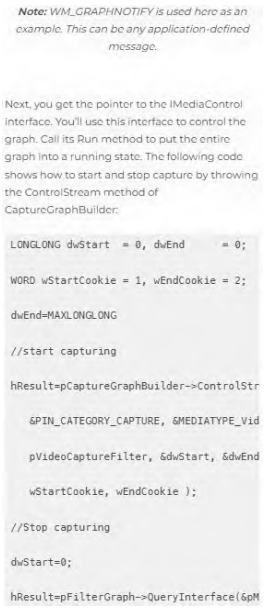
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

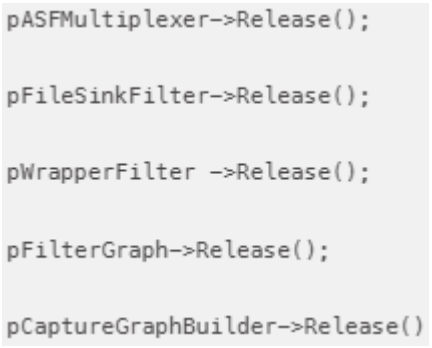
    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

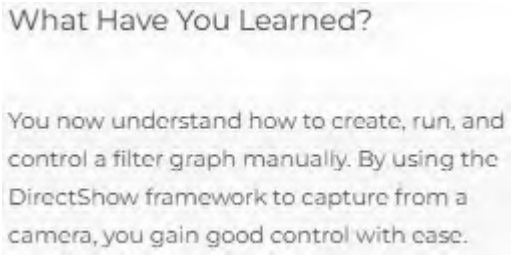
    &dwStart, &dwEnd, wStartCookie, wEndC
```

The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre]–1[b], 1[e], 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1i] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Camera Driver (DMO)<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1549 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1551 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

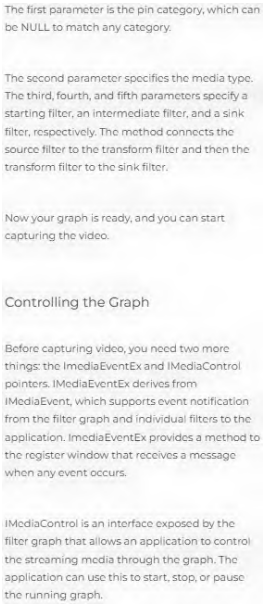
```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10  The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

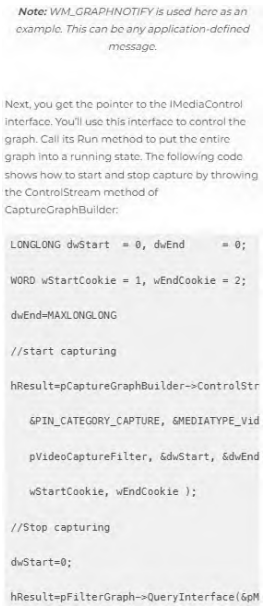
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

    &dwStart, &dwEnd, wStartCookie, wEndC
```

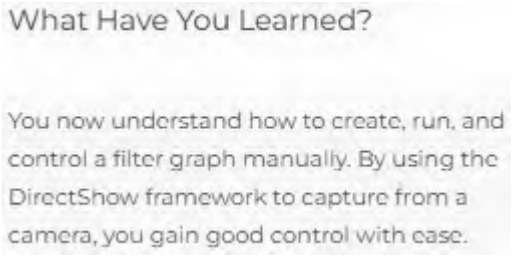
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1562 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre]–1[b], 1[e], 1[g], 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1j] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1567 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

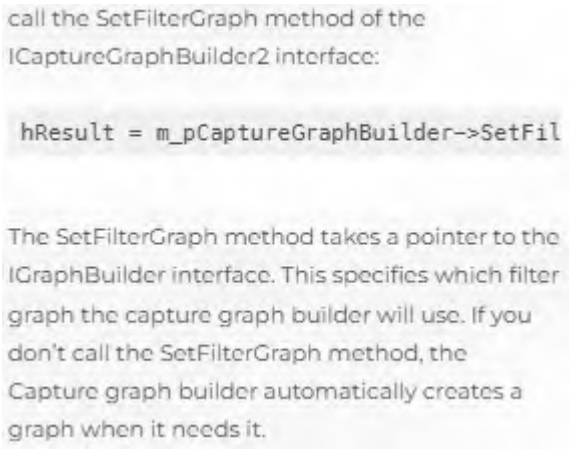
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1572 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1574 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |

The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

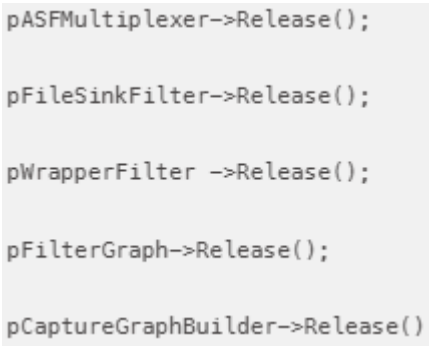
    &dwStart, &dwEnd, wStartCookie, wEndC
```

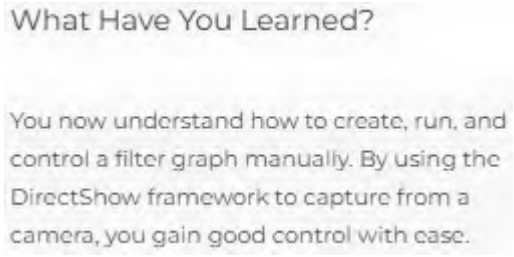
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1578 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[e], 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1k] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1583 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

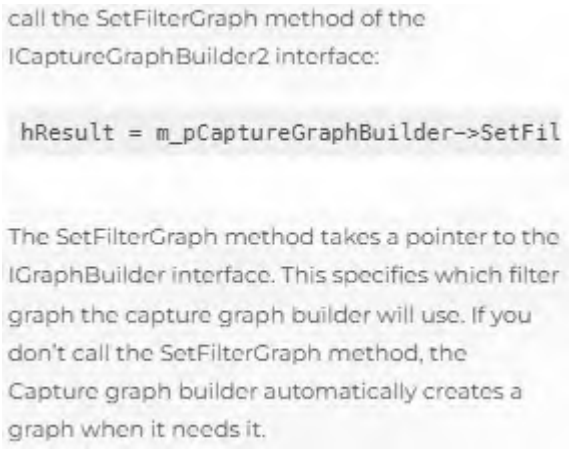
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

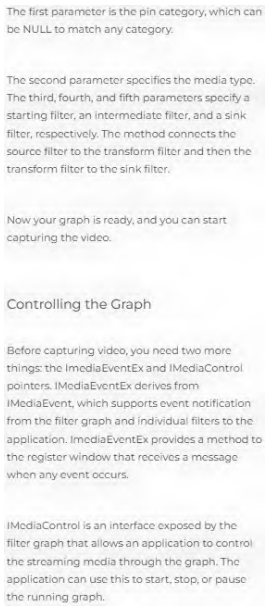
```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10  The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

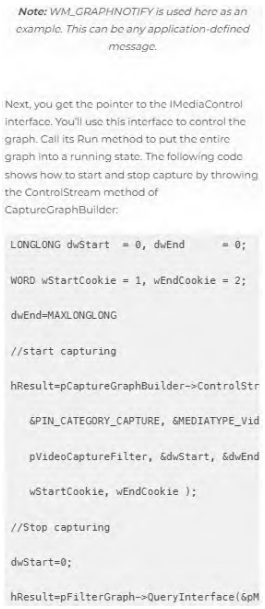
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

    &dwStart, &dwEnd, wStartCookie, wEndC
```

The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

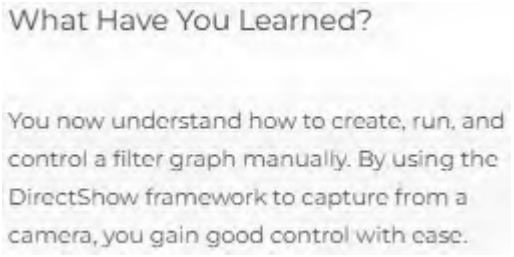
Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1594 of 1772

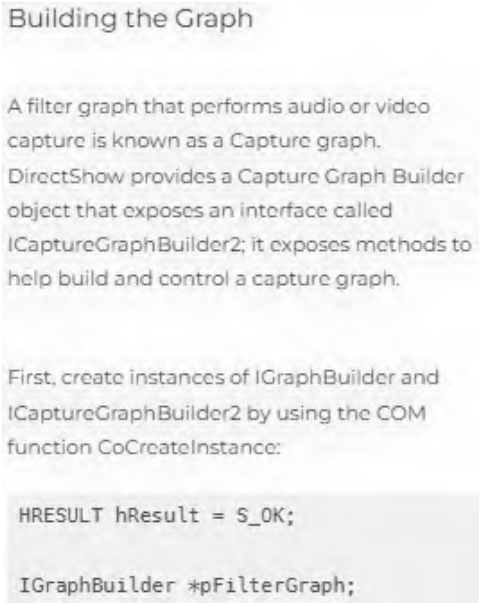
| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1595 of 1772

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1l] a server, remote from the user's mobile device, comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the server to perform operations including: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a server, remote from the user's mobile device, comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the server to perform operations including." For example:<br>*See also* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1m] receiving the digital image of the check transmitted by the user's mobile device, and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "receiving the digital image of the check transmitted by the user's mobile device." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3<br>  <br>Building the Graph<br>A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.<br>First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:<br>`HRESULT hResult = S_OK;`<br>`IGraphBuilder *pFilterGraph;` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1599 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

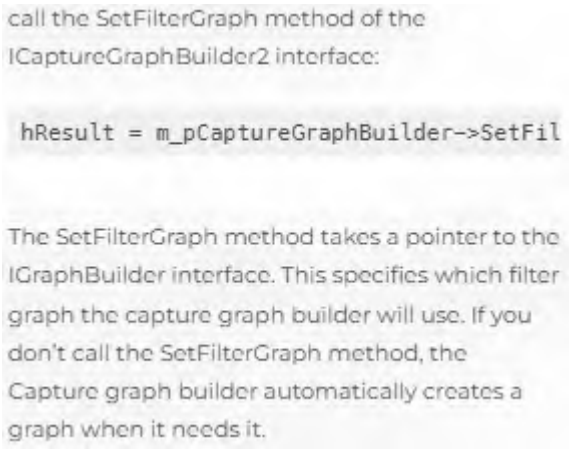
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1604 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

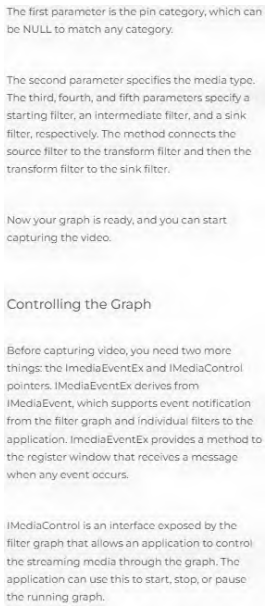
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10  The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

    &dwStart, &dwEnd, wStartCookie, wEndC
```

The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

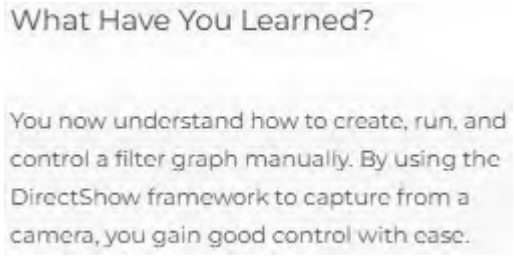
Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1610 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1611 of 1772

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[j], 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1n] processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, the user's mobile device receiving an Indication of a result of the server's processing before confirming that the mobile check deposit can go forward; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing the digital image of the check including performing optical character recognition on the check in the digital image and verifying that the digital image of the check is acceptable for mobile check deposit, the user's mobile device receiving an Indication of a result of the server's processing before confirming that the mobile check deposit can go forward." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br><br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br> |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1614 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1615 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

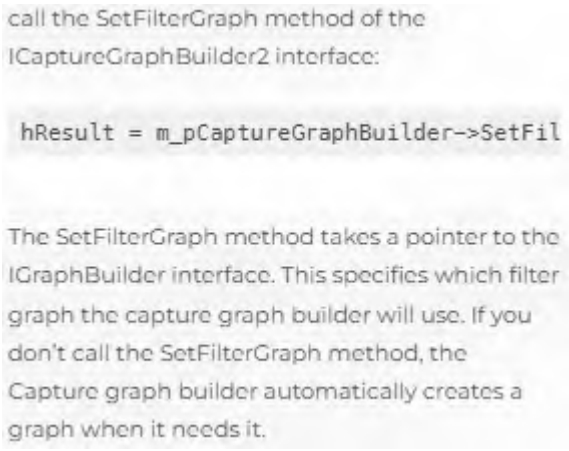
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

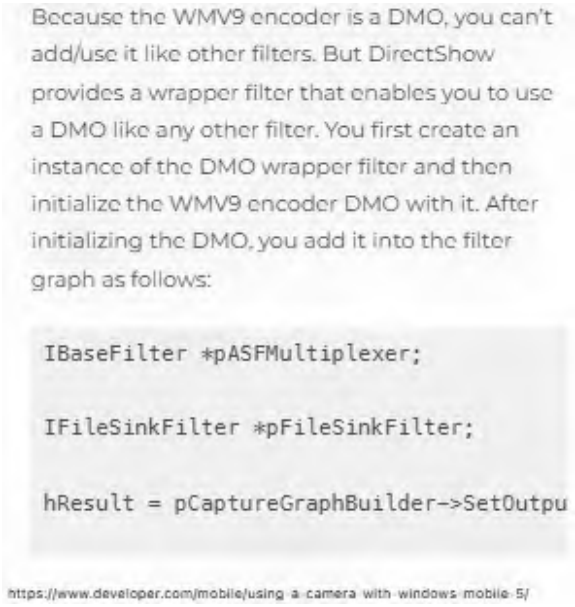
| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1620 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>  <br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

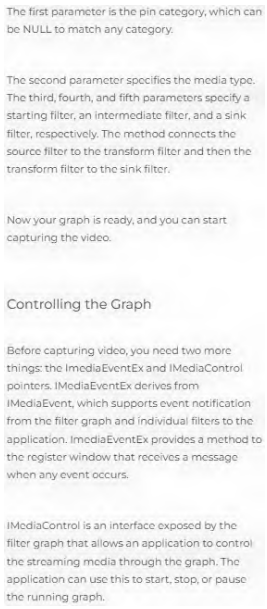
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10  The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

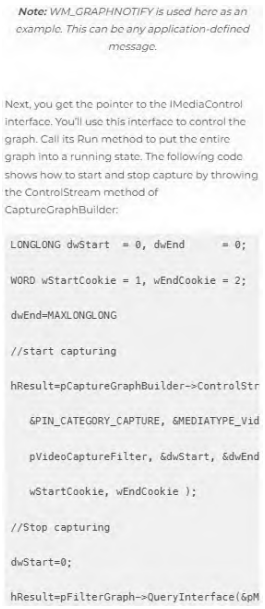
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

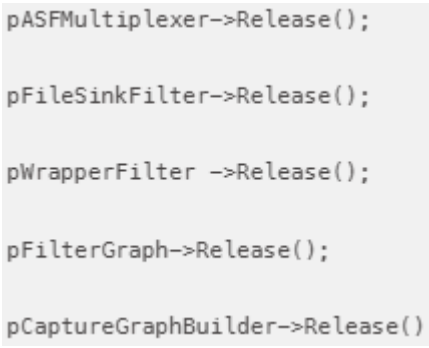
    &dwStart, &dwEnd, wStartCookie, wEndC
```

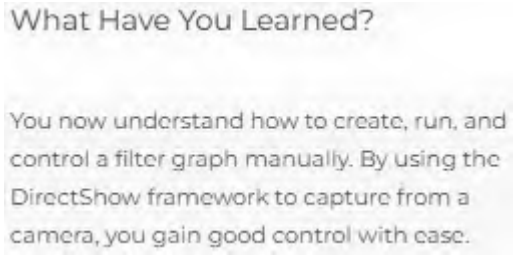
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1626 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[k], 1[l].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [1o] a computer comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the computer to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a computer comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the computer to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the digital image of the check is acceptable for mobile check deposit." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Camera Driver (DMO)<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1631 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

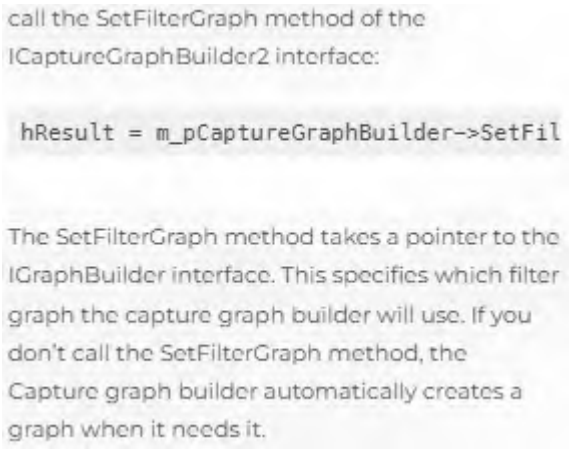
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1636 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1638 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |

The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1639 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

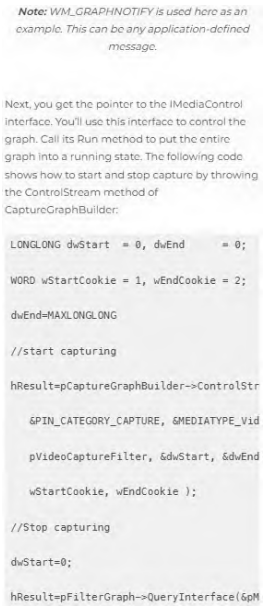
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

    &dwStart, &dwEnd, wStartCookie, wEndC
```

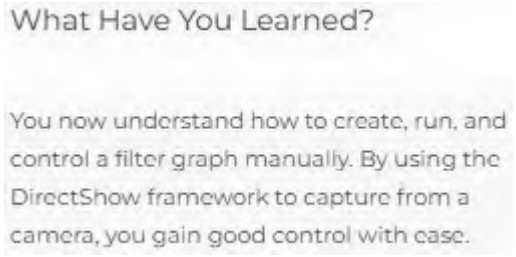
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1642 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre], 1[kj]–1[l], 1[n].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 2[pre]. The system of claim 1, wherein the downloaded app causes the user's mobile device to perform additional operations including: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 1, wherein the downloaded app causes the user's mobile device to perform additional operations including." For example:<br>*See* limitations 1[pre], 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [2a] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitations 1[pre]-1[b], 1[e], 1[g], 1[h], 2[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [2b] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br><br>*See* limitations 1[pre]-1[b], 1[e], 1[g]-1[i], 2[pre], 2[a].<br><br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 3. The system of claim 1, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 1, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br><br>Ranjan at page 1<br><br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br><br>Figure 1: Filter Graph for Video Capture |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at Page 1<br>  <br><br>Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3<br>Building the Graph<br>A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.<br>First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:<br>`HRESULT hResult = S_OK;`<br>`IGraphBuilder *pFilterGraph;` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1647 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1652 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

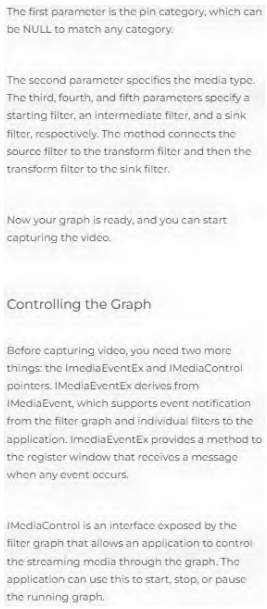
    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1654 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10  The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

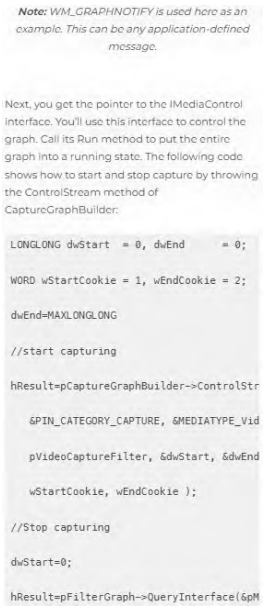
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

    &dwStart, &dwEnd, wStartCookie, wEndC
```

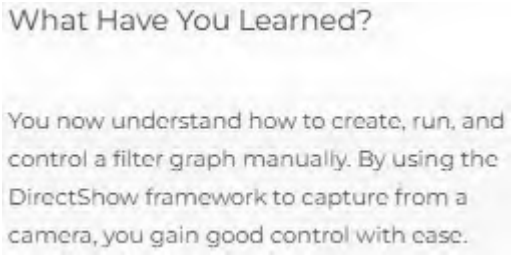
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.

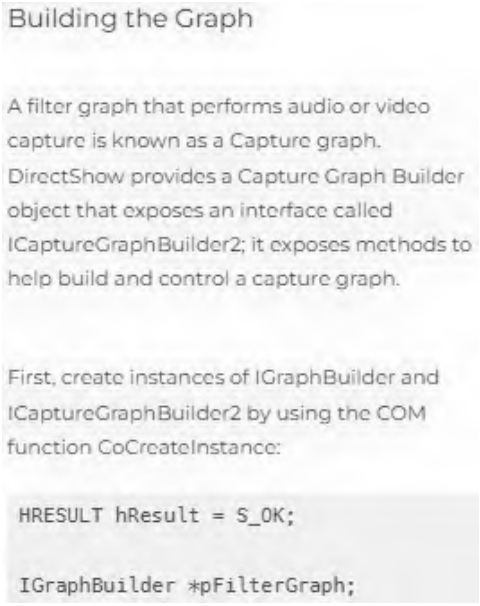

USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1658 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br>Ranjan at page 17<br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[pre]–1[b], 1[e], 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 4. The system of claim 3, wherein the plurality of monitoring criteria Includes a criterion based on magnetic ink character recognition (MICR) line detection. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 3, wherein the plurality of monitoring criteria Includes a criterion based on magnetic ink character recognition (MICR) line detection." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3<br>  <br>Building the Graph<br>A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.<br>First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:<br>`HRESULT hResult = S_OK;`<br>`IGraphBuilder *pFilterGraph;` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

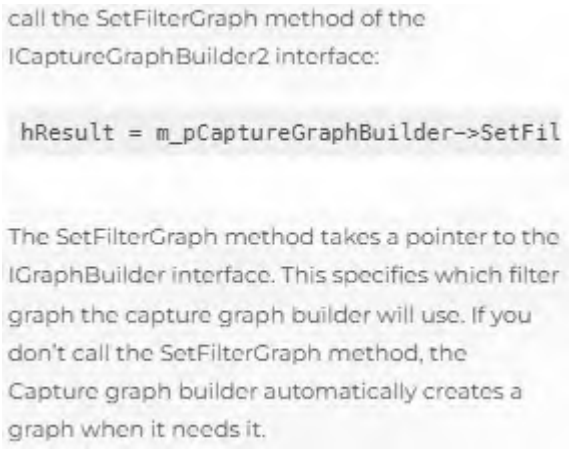
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1668 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

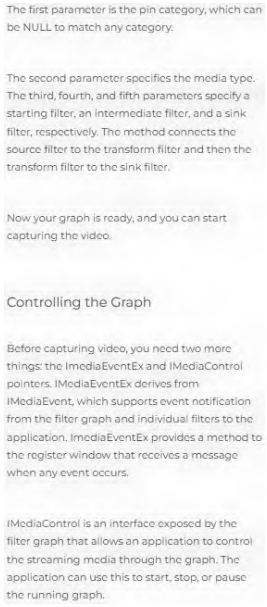
&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |



The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1671 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

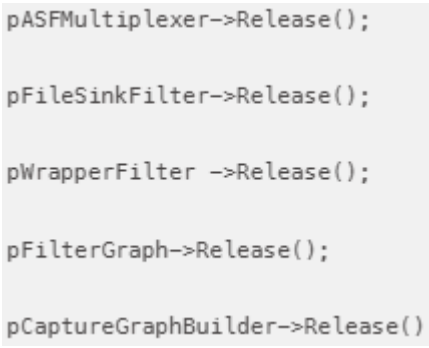
    &dwStart, &dwEnd, wStartCookie, wEndC
```

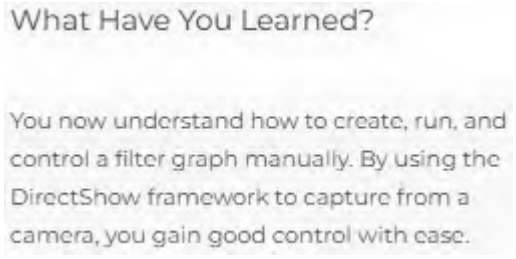
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1674 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* claim 3.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 5. The system of claim 3, wherein the plurality of monitoring criteria Includes criteria based on light brightness, skewing, and warping. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 3, wherein the plurality of monitoring criteria Includes criteria based on light brightness, skewing, and warping." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Capture<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1678 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1679 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1683 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1684 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1686 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |

The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

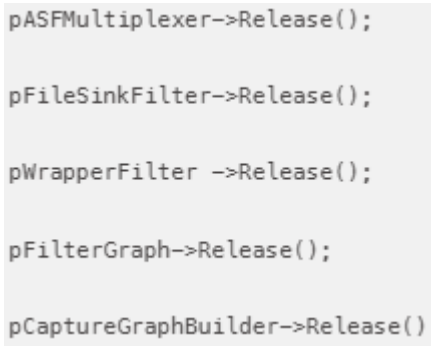
    &dwStart, &dwEnd, wStartCookie, wEndC
```

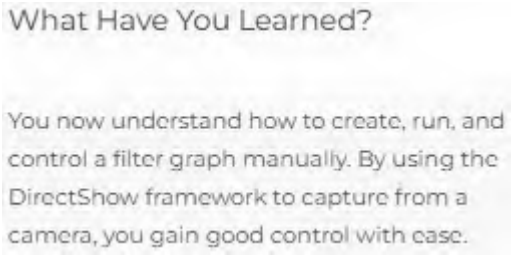
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1690 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* claim 3.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 6. The system of claim 1, wherein the confirming that the mobile check deposit can go forward is performed after the transmitting the digital image of the check from the user's mobile device. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 1, wherein the confirming that the mobile check deposit can go forward is performed after the transmitting the digital image of the check from the user's mobile device." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br><br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1694 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3<br>Building the Graph<br>A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.<br>First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:<br>`HRESULT hResult = S_OK;`<br>`IGraphBuilder *pFilterGraph;` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

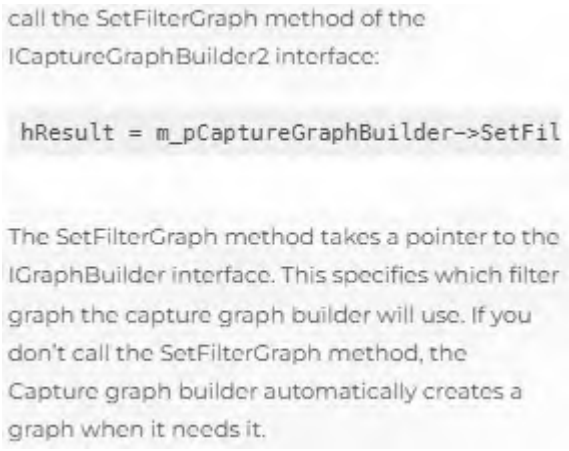
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1699 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1700 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1702 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10 |

The first parameter is the pin category, which can be NULL to match any category.

The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.

Now your graph is ready, and you can start capturing the video.

Controlling the Graph

Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.

IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

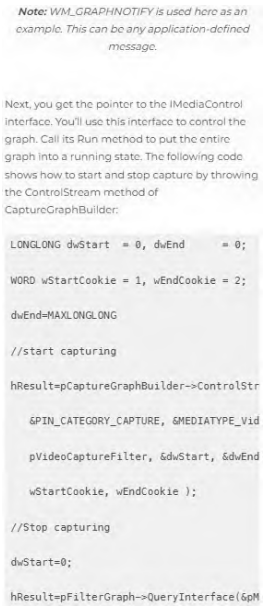
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

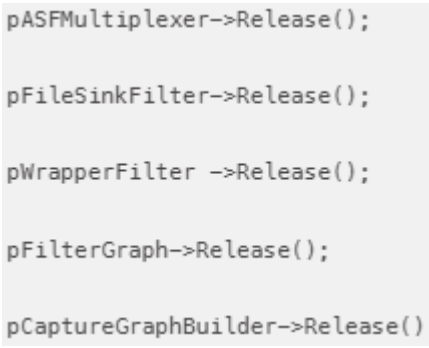
    &dwStart, &dwEnd, wStartCookie, wEndC
```

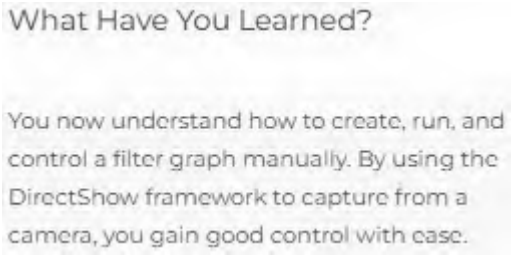
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1706 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[i], 1[k].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 7. The system of claim 1, wherein the digital image is a modified version of the captured video frame. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 1, wherein the digital image is a modified version of the captured video frame." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3 |

Building the Graph

A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.

First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:

```
HRESULT hResult = S_OK;

IGraphBuilder *pFilterGraph;
```


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1711 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

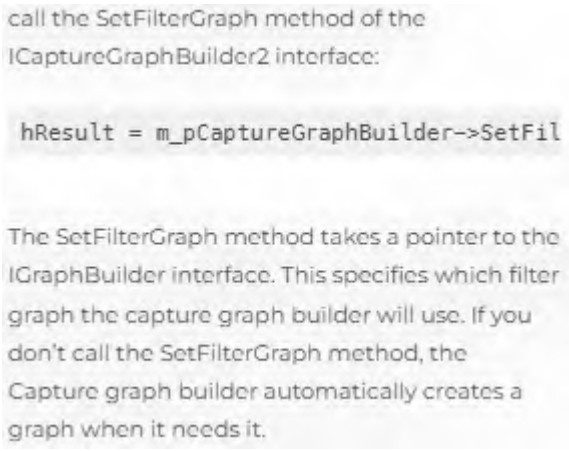
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1715 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

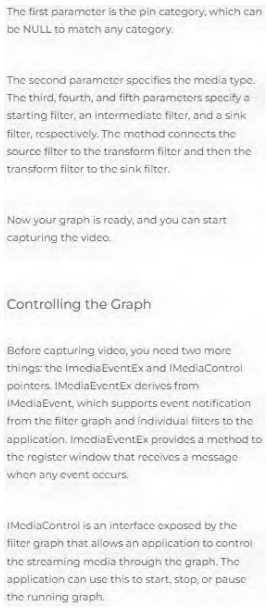
```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10  The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |

*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

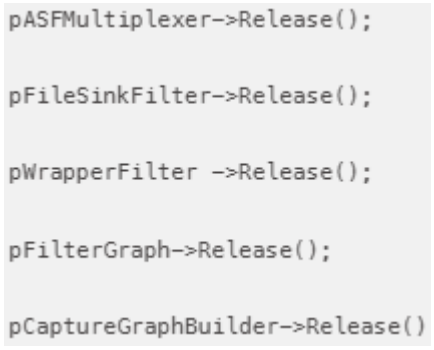
    &dwStart, &dwEnd, wStartCookie, wEndC
```

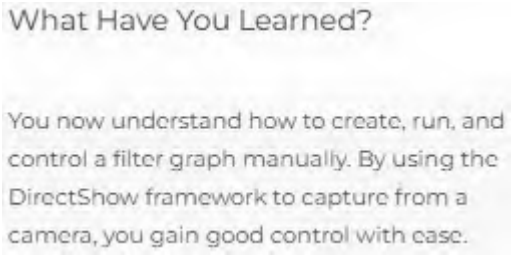
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1722 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitations 1[j], 1[m], 1[o].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 8. The system of claim 1, wherein the wireless network is a cellular network. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 1, wherein the wireless network is a cellular network." For example: |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Camera<br>Video Encoder<br>Multi-Plexer<br>File System<br>**Figure 1:** Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1726 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3<br>Building the Graph<br>A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.<br>First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:<br>`HRESULT hResult = S_OK;`<br>`IGraphBuilder *pFilterGraph;` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1727 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

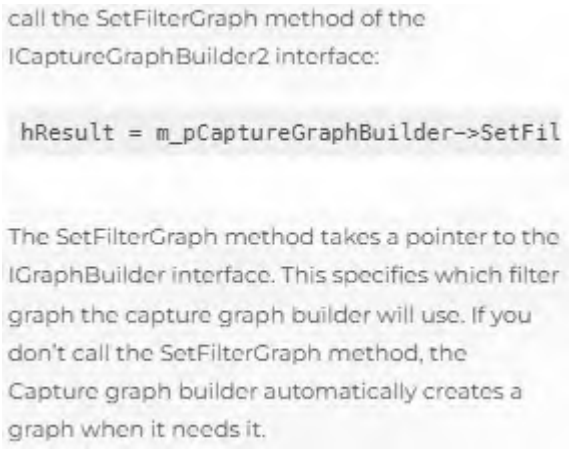
                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>  <br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>  <br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1732 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

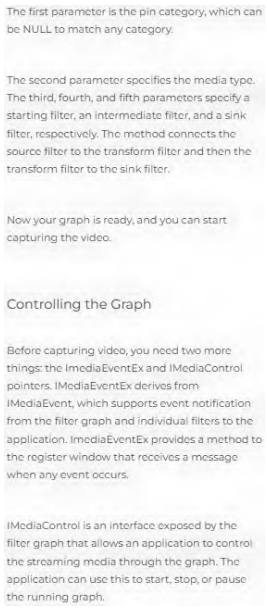
```
&MEDIASUBTYPE_Asf, T("test.asf"), &pA

&pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10  The first parameter is the pin category, which can be NULL to match any category.<br><br>The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter.<br><br>Now your graph is ready, and you can start capturing the video.<br><br>Controlling the Graph<br><br>Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs.<br><br>IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

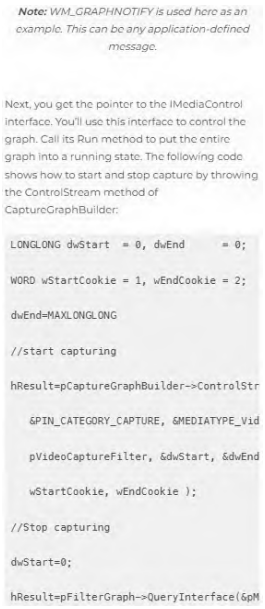
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

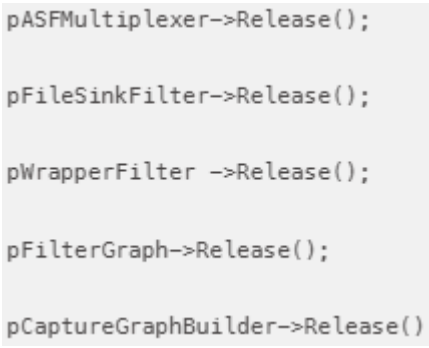
    &dwStart, &dwEnd, wStartCookie, wEndC
```

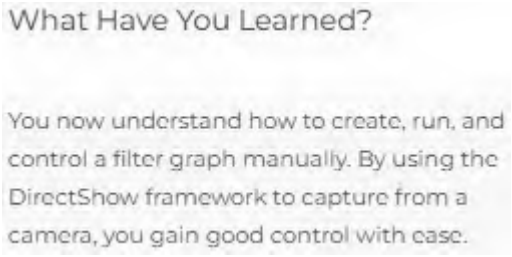
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1738 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See also* limitation 1[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 9[pre]. A system enabling mobile check deposit auto-capture functionality via video frame processing, the system comprising: | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] system enabling mobile check deposit auto-capture functionality via video frame processing, the system." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9a] a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the system being configured to perform operations including: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera, the system being configured to perform operations including." For example:<br>*See* limitations 1[a], 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [9b] providing Instructions to the user for capturing an image of a check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing Instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9c] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9d] overlaying an alignment guide on the live video of the check in the field of view of the camera; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "overlaying an alignment guide on the live video of the check in the field of view of the camera." For example:<br>*See* limitation 1[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9e] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [9f] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9g] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9h] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, the digital image of the check being the captured video frame or being based on the captured video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, the digital image of the check being the captured video frame or being based on the captured video frame." For example:<br>*See* limitation 1[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9i] processing the digital image of the check Including performing optical character recognition and verifying that the digital image of | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "processing the digital image of the check Including performing optical character recognition and verifying that the digital image of the check Is acceptable for mobile check deposit." For example:<br>*See* limitation 1[l], 1[n]. |

| Claims | Prior Art |
|---|---|
| the check Is acceptable for mobile check deposit; and | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9j] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check in the digital image, a magnetic ink character recognition (MICR) line on the check in the digital image is recognized, and an amount payable on the check in the digital image is validated." For example:<br>*See* limitation 1[k].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [9k] a bank computer programmed to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a bank computer programmed to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the check is submitted for mobile check deposit in the bank, confirming that the mobile check deposit can go forward, and verifying that the digital image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 1[o].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 10[pre]. The system of claim 9, wherein the system is configured to perform additional operations including: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 9, wherein the system is configured to perform additional operations including." For example:<br>*See* limitation 2[pre]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10a] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 2[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [10b] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 2[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 11. The system of claim 9, further comprising one or more computers programmed to perform operations comprising the monitoring, the determining, or both the monitoring and the determining. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 9, further comprising one or more computers programmed to perform operations comprising the monitoring, the determining, or both the monitoring and the determining." For example:<br>*See* limitations 9[a], 9[e], 9[f].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 12. The system of claim 9, wherein the at least one monitoring criterion is a plurality of | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 9, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, |

| Claims | Prior Art |
|---|---|
| monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* claim 3.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 13. The system of claim 12, wherein the plurality of monitoring criteria Includes criteria based on light brightness, skewing, and warping. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 12, wherein the plurality of monitoring criteria Includes criteria based on light brightness, skewing, and warping." For example:<br>*See* claim 5.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 14. The system of claim 9, wherein the confirming that the mobile check deposit can go forward is performed after the transmitting the digital image of the check from the user's mobile device. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 9, wherein the confirming that the mobile check deposit can go forward is performed after the transmitting the digital image of the check from the user's mobile device." For example:<br>*See* claim 6.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 15. The system of claim 14, wherein the user's mobile device receives an Indication of a result of the processing the digital image of the check before the confirming step. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 14, wherein the user's mobile device receives an Indication of a result of the processing the digital image of the check before the confirming step." For example:<br>*See* limitations 1[n], 9[i], 9[j]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 16. The system of claim 11, wherein the one or more computers include the bank computer. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 11, wherein the one or more computers include the bank computer." For example:<br>*See* limitations 9[a], 9[e], 9f], claim 11.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 17. The system of claim 9, wherein the wireless network is a cellular network. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 9, wherein the wireless network is a cellular network." For example:<br>*See* limitation 9[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 18[pre]. A system enabling mobile check deposit auto-capture functionality via video frame processing, the system comprising: | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] system enabling mobile check deposit auto-capture functionality via video frame processing, the system." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18a] a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing." For example:<br>*See* limitation 1[a]. |

| Claims | Prior Art |
|---|---|
| | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18b] a software abstraction layer, the software abstraction layer comprising Instructions that, when executed by the processor, control the camera; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a software abstraction layer, the software abstraction layer comprising Instructions that, when executed by the processor, control the camera." For example:<br>*See* limitation 1[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18c] a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, the downloaded app comprising Instructions that, when executed by the processor, cause the user's mobile device to perform operations including: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, the downloaded app comprising Instructions that, when executed by the processor, cause the user's mobile device to perform operations including." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18d] providing Instructions to the user for capturing an image of a check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing Instructions to the user for capturing an image of a check." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18e] generating a live video of the check in a field of view of the camera, the downloaded app | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example: |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1747 of 1772

| Claims | Prior Art |
|---|---|
| causing the live video to be generated by controlling the camera through the software abstraction layer; | *See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18f] monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example:<br>*See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18g] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18h] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18i] using a wireless network, transmitting a digital image of the | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the |

| Claims | Prior Art |
|---|---|
| check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18j] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 1[k].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [18k] a computer comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the computer to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a computer comprising a processor coupled to a memory storing Instructions that, when executed by the processor, cause the computer to update a balance of an account at a financial institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit and the confirming that the mobile check deposit can go forward." For example:<br>*See* limitation 1[o].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| 19[pre]. The system of claim 18, wherein the downloaded app causes the user's mobile device to perform additional operations including: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 18, wherein the downloaded app causes the user's mobile device to perform additional operations including." For example:<br>*See* limitation 2[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [19a] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 2[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [19b] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 2[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 20. The system of claim 18, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 18, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* claim 3. |

| Claims | Prior Art |
|---|---|
| check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 21. The system of claim 20, wherein the plurality of monitoring criteria being based on magnetic ink character recognition (MICR) line detection and one or both of corner detection and edge detection. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 20, wherein the plurality of monitoring criteria being based on magnetic ink character recognition (MICR) line detection and one or both of corner detection and edge detection." For example:<br>Ranjan at page 1<br>  <br>A number of Windows Mobile 5.0 APIs (for example, SHCameraCapture) make it trivial for a mobile application developer to access a camera, but their ease of use comes at a price—flexibility. Most of the time, using the API directly would offer a solution, but sometimes you need more control and flexibility. That's where Microsoft's DirectShow framework comes in. This article shows how to use DirectShow to access a camera. It demonstrates how to build a filter graph manually and how to handle graph events in the application message handler. Having some prior knowledge of DirectShow and COM will be helpful, but it's not necessary.<br>Figure 1: Filter Graph for Video Capture<br>Ranjan at Page 1<br>  <br>Using a Camera with Windows Mobile 5 |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 2<br>The camera is the hardware component. For an application to interact with the camera, it would need to talk to its drivers. Next, the video capture filter enables an application to capture video. After capture, you encode the data using WMV9EncMediaObject, a DirectX Media Object (DMO). You can use a DMO inside a filter graph with the help of a DMO Wrapper filter. Next, the encoded video data needs to be multiplexed. You use a Windows Media ASF writer filter for this task. The ASF writer multiplexes the video data and writes it to an .asf file. With that, your filter graph is ready. Now, it's just a matter of running it. As you will see, building the graph is pretty easy too. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1752 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 3<br>Building the Graph<br>A filter graph that performs audio or video capture is known as a Capture graph. DirectShow provides a Capture Graph Builder object that exposes an interface called ICaptureGraphBuilder2; it exposes methods to help build and control a capture graph.<br>First, create instances of IGraphBuilder and ICaptureGraphBuilder2 by using the COM function CoCreateInstance:<br>`HRESULT hResult = S_OK;`<br>`IGraphBuilder *pFilterGraph;` |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1753 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 4 |

```
ICaptureGraphBuilder2 *pCaptureGraphBuil

hResult=CoCreateInstance(CLSID_FilterGra

                         IID_IGraphBuild

hResult=CoCreateInstance(CLSID_CaptureGr

                         CLSCTX_INPROC,

                         (void**)& pCapt
```

CoCreateInstance takes five parameters:

1. The first is a class ID.
2. The second decides whether the object created is part of an aggregator.
3. The third specifies the context in which the newly created object would run.
4. The fourth parameter is a reference to the identifier of the interface you will use to communicate with the object.
5. The last parameter is the address of the variable that receives the interface pointer requested.

Once you have created the IGraphBuilder and ICaptureGraphBuilder2 instances, you need to

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 5<br>Now, you're ready to create an instance of the video capture filter. The following code initializes a Video capture filter, the pointer of which is returned by the CoCreateInstance:<br>`IBaseFilter *pVideoCaptureFilter`<br>`hResult=CoCreateInstance(CLSID_VideoCapt`<br>`IID_IBaseFilter`<br>You then need to get a pointer to IPersistPropertyBag from the video capture filter. You use this pointer to set the capture device (in other words, the camera) that the capture filter will use, as follows:<br>`IPersistPropertyBag *pPropertyBag;`<br>`hResult=pVideoCaptureFilter->QueryInterf`<br>Ranjan at page 5<br>call the SetFilterGraph method of the ICaptureGraphBuilder2 interface:<br>`hResult = m_pCaptureGraphBuilder->SetFil`<br>The SetFilterGraph method takes a pointer to the IGraphBuilder interface. This specifies which filter graph the capture graph builder will use. If you don't call the SetFilterGraph method, the Capture graph builder automatically creates a graph when it needs it. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 6<br>  <br>Now, you need to get a handle on the camera you will use to capture video. You can enumerate the available camera devices by using the FindFirstDevice and FindNextDevice functions. You can have multiple cameras present on a device. (HTC Universal is one example.) To keep the code simple for this example, use FindFirstDevice to get the first available camera on the device as follows:<br>`DEVMGR_DEVICE_INFORMATION devInfo;`<br>`CComVariant  CamName;`<br>`CPropertyBag PropBag;`<br>`GUID guidCamera = { 0xCB998A05, 0x122C,`<br>`0x3E, 0x4D, 0x7E, 0x`<br>`devInfo.dwSize = sizeof(devInfo);`<br>`FindFirstDevice( DeviceSearchByGuid, &gu`<br>`CamName=devInfo.szLegacyName`<br>`PropBag.Write( _T("VCapName"), &CamName`<br>`pPropertyBag->Load( &PropBag, NULL );`<br>`hResult =pFilterGraph->AddFilter( pVideo`<br>`_T("Vi`<br>Ranjan at page 7<br>`pPropertyBag.Release()` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 7<br>In the code sample, CPropertyBag is a custom class that has implemented IPropertyBag. This object is needed to pass the capture device name to the filter. The string VCapName identifies the filter property for the name of the video capture device. Once you have set the capture device, you can add the Video capture filter to the filter graph. You use the AddFilter method of the graph manager for this. This method takes two parameters: the first is the pointer to the filter that is to be added, and the second is the name of the filter. The second parameter can be NULL; in this case, the filter graph manager generates a unique name for the filter. If you have provided a name that conflicts with some other filter, the manager will modify the name to make it unique.<br>Ranjan at page 7<br>Note the first parameter in the FindFirstDevice, DeviceSearchByGuid. It specifies the search type. Other options are DeviceSearchByLegacyName, DeviceSearchByDeviceName, and so forth. DeviceSearchByGuid is the most reliable way to find a capture device. The information regarding the device is returned in the DEVMGR_DEVICE_INFORMATION structure. You store the szLegacyName value in the CComVariant variable, and you need an object that has implemented IPropertyBag interface. |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>You then need to instantiate the WMV9 encoder:<br>`IBaseFilter *pVideoEncoder;`<br>`IDMOWrapperFilter *pWrapperFilter;`<br>`hResult=CoCreateInstance(CLSID_DMOWrappe`<br>`IID_IBaseFilter`<br>`hResult =pVideoEncoder->QueryInterface(`<br>`hResult =pWrapperFilter->Init( CLSID_CWM`<br>`DMOCATEGO`<br>`hResult=pFilterGraph->AddFilter( pVideoE` |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 8<br>Because the WMV9 encoder is a DMO, you can't add/use it like other filters. But DirectShow provides a wrapper filter that enables you to use a DMO like any other filter. You first create an instance of the DMO wrapper filter and then initialize the WMV9 encoder DMO with it. After initializing the DMO, you add it into the filter graph as follows:<br>`IBaseFilter *pASFMultiplexer;`<br>`IFileSinkFilter *pFileSinkFilter;`<br>`hResult = pCaptureGraphBuilder->SetOutpu`<br>https://www.developer.com/mobile/using-a-camera-with-windows-mobile-5/ |

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 9 |

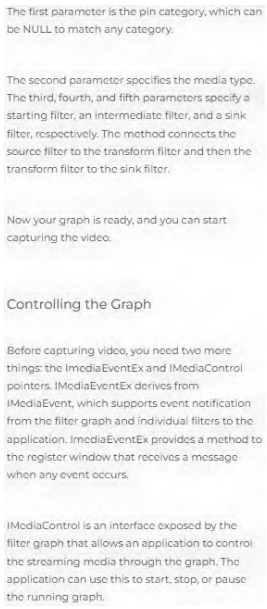
```
    &MEDIASUBTYPE_Asf, T("test.asf"), &pA

    &pFileSinkFilter );
```

You have added the source and the transform filter in the filter graph, so the last thing remaining is adding a sink filter in the graph. For this, you call the SetOutputFileName method of ICaptureGraphBuilder2. The first parameter is a media subtype; the second parameter is the name of the file in which you want to save the video; the third parameter is the address of a pointer that receives the multiplexer's interface; and the fourth parameter receives the file writers' interface.

With that, your filter graph is ready. All you need to do is connect the source filter, encoder, and multiplexer. You can achieve this by using the RenderStream method of the graph builder, as follows:

```
hResult = pCaptureGraphBuilder->RenderSt
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 10  The first parameter is the pin category, which can be NULL to match any category. The second parameter specifies the media type. The third, fourth, and fifth parameters specify a starting filter, an intermediate filter, and a sink filter, respectively. The method connects the source filter to the transform filter and then the transform filter to the sink filter. Now your graph is ready, and you can start capturing the video. Controlling the Graph Before capturing video, you need two more things: the IMediaEventEx and IMediaControl pointers. IMediaEventEx derives from IMediaEvent, which supports event notification from the filter graph and individual filters to the application. IMediaEventEx provides a method to the register window that receives a message when any event occurs. IMediaControl is an interface exposed by the filter graph that allows an application to control the streaming media through the graph. The application can use this to start, stop, or pause the running graph. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1761 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 11 |

The following code sample first queries the filter graph for its IMediaEventEx interface. Once it gets the pointer to the IMediaEventEx interface, it then calls its method SetNotifyWindow, passing it the handle to the window that handles the message. The second parameter is the message that will be passed as notification to the Windows message handler. The third parameter is the instance data (this can be 0).

```
IMediaEventEx *pMediaEvent;

IMediaControl *pMediaControl;

#define WM_GRAPHNOTIFY WM_APP+1

hResult =pFilterGraph->QueryInterface( I

                                       &

hResult =pMediaEvent->SetNotifyWindow((O

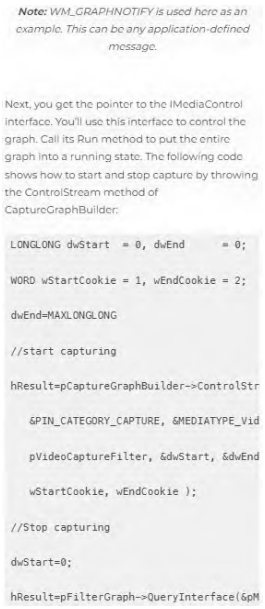
                                      WM

hResult=pFilterGraph->QueryInterface(&pM

hResult =pMediaControl->Run();
```

When an event occurs, DirectShow will send WM_GRAPHNOTIFY to the specified windows.

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 12 |



*Note: WM_GRAPHNOTIFY is used here as an example. This can be any application-defined message.*

Next, you get the pointer to the IMediaControl interface. You'll use this interface to control the graph. Call its Run method to put the entire graph into a running state. The following code shows how to start and stop capture by throwing the ControlStream method of CaptureGraphBuilder:

```
LONGLONG dwStart  = 0, dwEnd      = 0;
WORD wStartCookie = 1, wEndCookie = 2;
dwEnd=MAXLONGLONG
//start capturing
hResult=pCaptureGraphBuilder->ControlStr
   &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid
   pVideoCaptureFilter, &dwStart, &dwEnd
   wStartCookie, wEndCookie );
//Stop capturing
dwStart=0;
hResult=pFilterGraph->QueryInterface(&pM
```

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 13 |

```
hResult=pMediaSeeking->GetCurrentPositio

hResult= pCaptureGraphBuilder->ControlSt

    &PIN_CATEGORY_CAPTURE, &MEDIATYPE_Vid

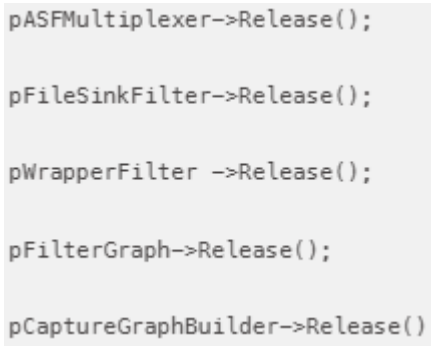
    &dwStart, &dwEnd, wStartCookie, wEndC
```

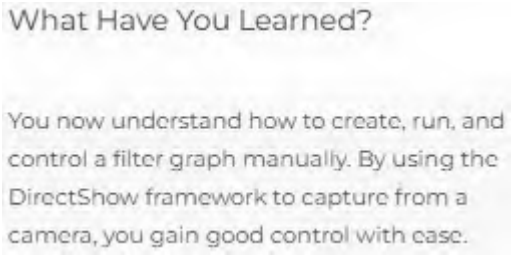
The code uses the search criteria supplied in the method call to locate an output pin on the capture filter. ControlStream enables an application to control streams without it needing to enumerate filters and pins in the graph.

Start and End specify the start and stop times (MAX_LONGLONG is the largest possible reference time value). When you start, the End is set to MAXLONLONG. When you want to stop, you first get the current position of the stream by using the GetCurrentPosition method of the IMediaSeeking interface. You then call the ControlStream method with Start set at 0 and End set at the current position.

You now have the graph ready and running. You can start using it to capture and save in an .asf file.


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1764 of 1772

| Claims | Prior Art |
| --- | --- |
| | Ranjan at page 16<br>  <br>You handle the WM_GRAPHNOTIFY message in the windows handler. DirectShow sends this message to the application when any event arises. The application calls a user-defined method to process the events. The GetEvent method of the IMediaEvent interface retrieves the event code and two event parameters from the queue.<br><br>Because the message loop and event notification are asynchronous, the queue might hold more then one event. Hence, the GetEvent code is called in a loop until it returns a failure code. Also, whenever you call GetEvent, it's important to call FreeEvent to free the resource associated with the event parameter. And, being the good programmer that you are, you won't forget to release the resources afterwards, will you? Call Release on every object that you have created, as follows:<br>`PVideoCaptureFilter->Release ();`<br>`pVideoEncoder->Release ();`<br>`pMediaEvent ->Release();`<br>`pMediaSeeking ->Release();`<br><br>Ranjan at page 17<br>  <br>`pASFMultiplexer->Release();`<br>`pFileSinkFilter->Release();`<br>`pWrapperFilter ->Release();`<br>`pFilterGraph->Release();`<br>`pCaptureGraphBuilder->Release()` |

| Claims | Prior Art |
|---|---|
| | Ranjan at page 17<br>  <br><br>*See* claims 3, 4, 20.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 22. The system of claim 20, wherein the plurality of monitoring criteria Includes criteria based on light brightness, skewing, and warping. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 20, wherein the plurality of monitoring criteria Includes criteria based on light brightness, skewing, and warping." For example:<br>*See* claim 5.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 23. The system of claim 18, wherein the confirming that the mobile check deposit can go forward is performed after the transmitting the digital image of the check from the user's mobile device. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 18, wherein the confirming that the mobile check deposit can go forward is performed after the transmitting the digital image of the check from the user's mobile device." For example:<br>*See* claim 6.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 24. The system of claim 18, wherein the digital image is a | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 18, wherein the digital image is a modified version of the captured video frame." For example: |

| Claims | Prior Art |
|---|---|
| modified version of the captured video frame. | *See* claim 7.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 25. The system of claim 18, wherein the wireless network is a cellular network. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 18, wherein the wireless network is a cellular network." For example:<br>*See* claim 8.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 26[pre]. A system enabling mobile check deposit auto-capture functionality via video frame processing, the system comprising: | To the extent the preamble is limiting, Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[a] system enabling mobile check deposit auto-capture functionality via video frame processing, the system." For example:<br>*See* limitation 1[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26a] a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a mobile device of a user, the user's mobile device including a camera, a processor, and a memory coupled to the processor, the memory storing." For example:<br>*See* limitation 1[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26b] a software abstraction layer, the software abstraction layer comprising Instructions that, when | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a software abstraction layer, the software abstraction layer comprising Instructions that, when executed by the processor, control the camera." For example:<br>*See* limitation 1[b]. |


USAA Exhibit 2004
Regions Bank v. USAA
IPR2025-01356, Page 1767 of 1772

| Claims | Prior Art |
|---|---|
| executed by the processor, control the camera; and | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26c] a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, the downloaded app comprising Instructions that, when executed by the processor, cause the user's mobile device to perform operations including: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a downloaded app, the downloaded app associated with a bank to control check deposit by controlling the camera through the software abstraction layer, the downloaded app comprising Instructions that, when executed by the processor, cause the user's mobile device to perform operations including." For example:<br>*See* limitation 1[c].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26d] providing Instructions to the user for capturing a check; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "providing Instructions to the user for capturing a check." For example:<br>*See* limitation 1[d].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26e] generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "generating a live video of the check in a field of view of the camera, the downloaded app causing the live video to be generated by controlling the camera through the software abstraction layer." For example:<br>*See* limitation 1[e].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26f] monitoring the live video of the check in the field of view of | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "monitoring the live video of the check in the field of view of the camera with respect to at least one monitoring criterion." For example: |

| Claims | Prior Art |
|---|---|
| the camera with respect to at least one monitoring criterion; | *See* limitation 1[g].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26g] determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "determining that a video frame of the live video of the check satisfies the at least one monitoring criterion and then automatically capturing the video frame." For example:<br>*See* limitation 1[h].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26h] when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "when the video frame of the live video of the check in the field of view of the camera is determined to satisfy the at least one monitoring criterion, providing feedback advising the user that the check has been captured." For example:<br>*See* limitation 1[i].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26i] using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "using a wireless network, transmitting a digital image of the check from the user's mobile device and submitting the check for mobile check deposit in the bank after automatically capturing the video frame, wherein the digital image is the captured video frame or is based on the captured video frame." For example:<br>*See* limitation 1[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |

| Claims | Prior Art |
|---|---|
| [26j] confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated; | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "confirming that the mobile check deposit can go forward after optical character recognition is performed on the check, a magnetic ink character recognition (MICR) line on the check is recognized, and an amount payable on the check is validated." For example:<br>*See* limitation 18[j].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26k] the system configured to perform image processing operations on an image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "the system configured to perform image processing operations on an image of the check, the image processing operations including performing optical character recognition and verifying that the image of the check is acceptable for mobile check deposit, wherein the image of the check is the digital image transmitted by the user's mobile device or is based on the digital image transmitted by the user's mobile device." For example:<br>*See* limitations 1[j], 1[n].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [26l] a computer programmed to update a balance of an account at a financial Institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "a computer programmed to update a balance of an account at a financial Institution of the user to reflect an amount of funds associated with the check submitted for mobile check deposit by the user's mobile device, after the submitting the check for mobile check deposit in the bank, the confirming that the mobile check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit." For example:<br>*See* limitation 1[o]. |

| Claims | Prior Art |
|---|---|
| check deposit can go forward, and the verifying that the image of the check is acceptable for mobile check deposit. | This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 27[pre]. The system of claim 26, wherein the downloaded app causes the user's mobile device to perform additional operations including: | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 26, wherein the downloaded app causes the user's mobile device to perform additional operations including." For example:<br>*See* limitation 2[pre].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [27a] before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion; and | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "before determining that the video frame of the live video of the check satisfies the at least one monitoring criterion, determining that a different video frame of the live video of the check does not satisfy the at least one monitoring criterion." For example:<br>*See* limitation 2[a].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| [27b] after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "after determining that the different video frame of the live video of the check does not satisfy the at least one monitoring criterion, providing feedback to the user." For example:<br>*See* limitation 2[b].<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 28. The system of claim 26, wherein the at least one monitoring criterion is a plurality of | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 26, wherein the at least one monitoring criterion is a plurality of monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, |

| Claims | Prior Art |
|---|---|
| monitoring criteria based on two or more selected from a group consisting of light contrast, light brightness, check positioning, check dimensions, character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection. | character spacing, skewing, warping, corner detection, edge detection, and magnetic ink character recognition (MICR) line detection." For example:<br>*See* claim 3.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 29. The system of claim 28, wherein the plurality of monitoring criteria Includes a criterion based on magnetic ink character recognition (MICR) line detection. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 28, wherein the plurality of monitoring criteria Includes a criterion based on magnetic ink character recognition (MICR) line detection." For example:<br>*See* claim 4.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |
| 30. The system of claim 26, wherein the wireless network is a cellular network. | Ranjan discloses and/or renders obvious, at least under USAA's apparent infringement theories, "[t]he system of claim 26, wherein the wireless network is a cellular network." For example:<br>*See* claim 8.<br>This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Ranjan alone; (2) Ranjan in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Regions' other invalidity charts and cover pleading. |