

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ALERTUS TECHNOLOGIES, LLC
Petitioner,

v.

DESKTOP ALERT, INC.
Patent Owner.

Case IPR2024-00363
Patent No. 9,172,765

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 9,172,765**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES AND PAYMENT OF FEES.....	2
III.	GROUND FOR STANDING.....	3
IV.	THE '765 PATENT.....	3
	A. Related Application.....	3
	B. Priority Date	3
	C. Relevant Prosecution History	3
V.	LEVEL OF ORDINARY SKILL IN THE ART	4
VI.	CLAIM CONSTRUCTION	5
	A. Terms Lacking Antecedent Basis	5
	1. “Said Memory”	6
	2. “Its Memory”	6
	3. “Said Processor”	6
	B. Terms in Claims 10 and 12	7
	1. “Get Alert Messages”	7
	2. “Said Client Displays, Display Said Alter Messages”	8
VII.	REQUESTED RELIEF	8
VIII.	OVERVIEW OF CHALLENGES	8
	A. Statutory Grounds for Challenges.....	8
	B. Prior-Art Status	9
IX.	DISCRETIONARY DENIAL IS UNWARRANTED.....	11
	A. <i>Fintiv</i> Weighs Against Denial.....	11
	B. Section 325(d) Denial is Unwarranted.....	15
X.	DETAILED EXPLANATION OF GROUNDS.....	15
	A. Ground 1: <i>Dillon</i> -AAPA Renders Obvious Claims 1-13.....	15
	1. Overview of <i>Dillon</i>	15
	2. Overview of AAPA	17

3.	Reasons to Combine <i>Dillon</i> and AAPA	20
4.	Claim 1	24
a)	[1.1] An improved method for the dissemination of an alert message to a plurality of personal computers using a network communications protocol	24
b)	[1.2] said computers comprising clients, each said clients subscribing to a central server, said server comprising an alert message cache, said server adapted to receive and store alert messages, and	27
c)	[1.3] and said server adapted to generate and store alert ID numbers, ...	29
d)	[1.4] said method comprising a first triggering step comprises the use of a HTTP packet comprising a request for a Boolean value relating to whether there are new alert messages and said packet comprises approximately 20 bytes	30
e)	[1.5] a second step wherein said client contacts the central server and requests the value of said alert ID number in said memory, said value reflecting the number of alert messages received by said central server	34
f)	[1.6] a third step wherein said server transmits said value to said client	36
g)	[1.7] a fourth step wherein said client compares a stored value at the client with the value of the transmitted alert ID number, and.....	37
h)	[1.8] if said transmitted value exceeds said stored value, a fifth step wherein said client transmits a second message to said server and requests for the server to transmit all unread alert messages in a GetUnreadAlerts request from said server that are specific to said client, and	38

i)	[1.9] if there are unread alert messages specific to said client in said server, a sixth step wherein said server sends to client all unread alert messages specific to the client, and	40
j)	[1.10] a further step wherein said client updates the stored value in its memory to reflect the number of additional unread alert messages transmitted from said server to said client in response to said second message, and.....	41
k)	[1.11] wherein, upon the reception of said unread alert messages in said sixth step, said alert messages are in instant message format and said client opens a window on a display panel to display said alert messages,.....	42
l)	[1.12] wherein said communications protocol comprises protocols for the transmission of messages over the internet.....	44
5.	Claim 2.....	45
6.	Claim 3.....	46
7.	Claims 4 and 5	47
8.	Claim 6, 7, and 8.....	49
9.	Claim 9.....	49
10.	Claim 10.....	50
a)	[10.1] A system for providing alert messages to a plurality of clients using a polling-based network, said system comprising a plurality of clients,.....	50
b)	[10.2] at least one server in communication with said clients using a network connection,.....	52
c)	[10.3] said central server provided with input devices to receive alert messages, and a database, said database for storing said alert messages,.....	54
d)	[10.4] and said processor having programming instructions to create and store an alert ID	

	value[s] corresponding to alert messages stored in said database, and	56
e)	[10.5] said clients adapted to communicate with said server using an RPC communications protocol	58
f)	[10.6] wherein said protocol further comprises a first triggering step comprises the use of a HTTP packet comprising a request for a Boolean value relating to whether there are new alert messages and said packet comprises approximately 20 bytes, and.....	59
g)	[10.7] further communication from said client to said server requesting a Get Alert Messages using a SOAP communications protocol and.....	59
h)	[10.8] wherein said client is transmitted an Alert ID message value from said server to said client, and.....	60
i)	[10.9] if said Alert ID message value matches a stored message value, the client will further request from said server any alert messages to which it subscribes, and in response to said requests, said server transmits any unread alert messages and	60
j)	[10.10] said unread alert message are in instant message format and upon the client's reception of said unread alert messages said client opens a window on a display panel at said client, and	62
k)	[10.11] wherein said communications protocol comprises protocols for the transmission of messages over the internet.....	62
11.	Claim 11.....	63
12.	Claim 12.....	64
13.	Claim 13.....	65
B.	Ground 2: <i>Dillon-Brady</i> -AAPA Render Obvious Claims 1- 13	65
1.	Overview of <i>Brady</i>	65

2.	Reasons to Combine <i>Dillon</i> -AAPA with <i>Brady</i>	66
3.	Claims 1-13.....	67
C.	Ground 3: <i>Dillon-Douglas</i> -AAPA Render Obvious Claims 1-13	68
1.	Overview of <i>Douglas</i>	68
2.	Reasons to Combine <i>Dillon</i> -AAPA with <i>Douglas</i>	68
3.	Claims 1-13.....	70
D.	Ground 4: <i>Dillon-Brady-Douglas</i> -AAPA Render Obvious Claims 1-13	70
1.	Reasons to Combine <i>Dillon-Brady</i> -AAPA with <i>Douglas</i>	70
2.	Claims 1-13.....	71
XI.	CONCLUSION.....	72
XII.	CERTIFICATE OF COMPLIANCE	73
XIII.	CLAIMS APPENDIX (U.S. PAT. NO. 9,172,765).....	74
XIV.	CERTIFICATE OF SERVICE	79

PETITIONER’S EXHIBIT LIST¹

EX-1001	U.S. Patent No. 9,172,765 (“the ’765 Patent”)
EX-1002	Prosecution History of U.S. Patent No. 9,172,765
EX-1003	Declaration of William Henry Mangione-Smith in Support of Petition for <i>Inter Partes</i> Review of U.S. Patent No. 9,172,765
EX-1004	CV of William Henry Mangione-Smith
EX-1005	U.S. Patent No. 6,067,561 (“ <i>Dillon</i> ”)
EX-1006	U.S. Patent Application Publication No. 2009/0164586A1 (“ <i>Douglas</i> ”)
EX-1007	U.S. Patent Application Publication No. 2005/0038791A1 (“ <i>Ven</i> ”)
EX-1008	U.S. Patent Application Publication No. 2010/0205243A1 (“ <i>Brady</i> ”)
EX-1009	U.S. Provisional Application No. 61/096,617 (“ <i>Brady</i> Provisional”)
EX-1010	Network Working Group, RFC 1945, Hypertext Transfer Protocol—HTTP/1.0 (May 1996) (available at https://www.rfc-editor.org/rfc/rfc1945.txt) (“RFC 1945”)
EX-1011	Network Working Group, RFC 2616, Hypertext Transfer Protocol – HTTP/1.1 (June 1999) (available at https://www.rfc-editor.org/rfc/rfc2616.txt) (“RFC 2616”)

¹ This Petition uses **bold font** when quoting claim language of the ’765 Patent and *italic font* for added emphasis and for the names of prior art references. Colored markings on reproductions of Exhibit figures are annotations.

EX-1012	Network Working Group, RFC 5322, Internet Message Format (October 2008) (available at https://www.rfc-editor.org/rfc/rfc5322.txt) (“RFC 5322”)
EX-1013	U.S. Patent No. 8,825,092 B2 (“ <i>Sennett</i> ”)
EX-1014	U.S. Patent Application Publication No. 2005/0039048A1 (“ <i>Tosey</i> ”)
EX-1015	U.S. Patent Application Publication No. 2008/0034114A1 (“ <i>Ducey</i> ”)
EX-1016	Microsoft Corporation, <i>Microsoft Computer Dictionary</i> (5th ed. 2002) (excerpts)
EX-1017	U.S. District Courts—Federal Court Management Statistics—Profiles—During the 12-Month Periods Ending September 30, 2018 Through 2023 (available at https://www.uscourts.gov/statistics/table/na/federal-court-management-statistics/2023/09/30-1)
EX-1018	United States District Court for the District of Maryland, Local Rules (Jul. 1, 2023) (available at https://www.mdd.uscourts.gov/sites/mdd/files/LocalRules.pdf)
EX-1019	Docket, <i>Desktop Alert Inc. v. Alertus Technologies, LLC</i> , No. 8:22-cv-03337-DLB (D. Md.) (printed Jan. 7, 2024)
EX-1020	First Amended Complaint for Patent Infringement, <i>Desktop Alert Inc. v. Alertus Technologies, LLC</i> , No. 8:22-cv-03337-DLB (D. Md. Jun. 2, 2023) ²

² For documents filed in the Litigation (EX-1020—EX-1022), this Petition references page numbers inserted to the bottom left-hand corner of the page (as opposed to page numbering in the filed document itself).

EX-1021	Defendant Alertus Technologies, LLC’s Motion to Dismiss, <i>Desktop Alert Inc. v. Alertus Technologies, LLC</i> , No. 8:22-cv-03337-DLB (D. Md. Jun. 22, 2023)
EX-1022	Plaintiff’s Opposition to Defendant’s Motion to Dismiss and, in the Alternative, Motion for Summary Judgment of No Invalidity Under 35 U.S.C. § 101, <i>Desktop Alert Inc. v. Alertus Technologies, LLC</i> , No. 8:22-cv-03337-DLB (D. Md. Jul. 6, 2023)
EX-1023	Declaration of David H. Crocker in Support of Petition for <i>Inter Partes</i> Review of U.S. Patent No. 9,172,765
EX-1024	CV of David H. Crocker
EX-1025	Claims Appendix: U.S. Patent No. 9,172,765

I. INTRODUCTION

Alertus Technologies, LLC (“Petitioner”) requests *inter partes* review (“IPR”) of claims 1-13 (“Challenged Claims”) of U.S. Patent No. 9,172,765 (“’765 Patent”) (EX-1001) assigned to Desktop Alert Inc. (“Patent Owner”).

The ’765 Patent describes its purported invention as an improvement over Patent Owner’s Desktop Alert system, which the specification admits is a “*prior art*,” “*conventional*,” “*legacy*” system, and based on “*well-known*,” “*commonly used*,” and “*widely-adopted*” polling architectures and protocols. *See* Section X.A.2. The admissions regarding the prior art Desktop Alert system and other aspects constitute applicant admissions regarding the prior art (“AAPA”). Rather than focusing on AAPA, the prosecution centered on cited art. In response to a final office action, for instance, the applicant attempted to distinguish the purported invention over the cited art by emphasizing implementation details: the format of a packet (*i.e.*, “**an HTTP packet comprising a request for a Boolean value**”), the size of the packet (*i.e.*, “**said packet comprises approximately 20 bytes**”) and the format of messages (*i.e.*, “**alert messages are in instant message format**”). *See* Section IV.C. But in parallel litigation the Patent Owner has admitted that HTTP (along with other claimed protocols) were “*known* computer communication protocols” and/or “*conventional*.” *See* Section X.A.2. Other aspects were either (a)

disclosed or suggested in prior art or (b) obvious modifications to a person of ordinary skill.

As this Petition demonstrates, there is nothing inventive in the Challenged Claims as they are rendered obvious by the prior art references in this Petition (none of which were before the Examiner) and AAPA. The Board should institute and cancel the Challenged Claims.

II. MANDATORY NOTICES AND PAYMENT OF FEES

Real Parties-in-Interest: Petitioner identifies the following real parties-in-interest: Alertus Technologies, LLC, Jason A. Volk, The Volk Family Delaware Dynasty Trust I, and The Volk Family Delaware Dynasty Trust II.

Related Matters: Patent Owner asserts the '765 Patent against Petitioner in *Desktop Alert, Inc. v. Alertus Technologies, LLC*, No. 8:22-cv-03337-DLB (D. Md. filed Dec. 22, 2022) ("Litigation") filed in the United States District Court for the District of Maryland ("Court"). EX-1019-EX-1020.

Counsel and Service Information:

- **Lead Counsel:** Brian M. Koide (Reg. No. 41,123).
- **Backup Counsel:** (1) William F. Nixon (Reg. No. 44,262) and (2) Ryan Pool (Reg. No. 64,615).
- **Service Information:** Millen, White, Zelano, & Branigan, P.C., 2200 Clarendon Boulevard, Suite 1400, Arlington, VA 22201, Tel: (703)

243-6333, Fax: (703) 246-6410, Email: AlertusPTAB@mwzb.com.

Petitioner consents to electronic service.

Payment of Fees: The USPTO is authorized to charge any fees due during this proceeding to Deposit Account No. 13-3402.

III. GROUNDS FOR STANDING

Petitioner certifies that the '765 Patent is available for IPR and Petitioner is not barred or estopped from requesting an IPR of the Challenged Claims.

IV. THE '765 PATENT

A. Related Application

The application that led to the issuance of the '765 Patent was filed as App. No. 13/067,878 on July 1, 2011 (EX-1002), and purports to claim priority to Provisional Patent Application No. 61/398,796, filed on July 1, 2010 ("the '796 Provisional"). EX-1001, Cover.

B. Priority Date

This Petition assumes (without making any concession in other proceedings) that the Challenged Claims of the '765 Patent have a priority date of July 1, 2010, the filing date of the '796 Provisional.

C. Relevant Prosecution History

In a Final Office Action, the Examiner rejected all pending claims over the prior art of record. EX-1002 at 161-174. In response, the Applicant amended both

independent claims to include two limitations to pending independent claims: (1) “a first triggering step comprises the use of a HTTP packet comprising a request for a Boolean value relating whether there are new alert messages to and said packet comprises approximately 20 bytes” (“HTTP Packet Limitation”) and (2) “upon the reception of said unread alert messages in said sixth step, said alert messages are in instant message format and said client opens a window on a display panel to display said alert messages” (“Instant Message Limitation”). *Id.* at 145-48, 150. The Applicant emphasized both the HTTP Packet Limitation and Instant Message Limitation as distinguishable over the prior art before the Examiner. *Id.* at 152-153. After an Examiner interview (*Id.* at 110), the Examiner issued a Notice of Allowability, which included an Examiner Amendment (adding the word “to” to address grammatical issues). *Id.* at 103-109. The Examiner, however, never cited any of the prior art references that formulate the basis for this Petition. *See generally id.* Nor did the Examiner articulate any application of the AAPA. *Id.*

V. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art (“POSITA”) as of July 1, 2010 would have (i) a bachelor’s degree or equivalent in electrical engineering, computer engineering, network engineering, computer science, and/or a related field and (ii) 1-3 years of experience relating to computer and networking systems and/or methods including systems and methods for messaging. A person with less formal education

but more experience, or more formal education but less experience, could have also met the relevant standard. EX-1003, ¶¶25-26.

VI. CLAIM CONSTRUCTION

The claim terms below are construed “in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. §42.100(b). Terms not specifically construed below have their plain and ordinary meaning to a POSITA.

A. Terms Lacking Antecedent Basis

Terms in this section lack antecedent basis. Lack of antecedent basis, however, does not render a claim *per se* indefinite. *Energizer Holdings Inc. v. United States ITC*, 435 F.3d 1366, 1370-71 (Fed. Cir 2006). For the purposes of this Petition, Petitioner assumes the following constructions.³

³ The constructions assumed in this Petition are for the purposes of this Petition only. Petitioner reserves the right to argue indefiniteness, lack of written description, and other invalidity defenses in other proceedings (including the Litigation) for these and any other terms.

1. “Said Memory”

Claim 1 recites “**said client contacts the central server and requests the value of said alert ID number in said memory.**” EX-1001, 12:48-51. While claim 1 lacks antecedent basis for “**memory**,” claim 1 does have a prior reference to “**said server comprising an alert message cache.**” *Id.*, 12:40-41. This Petition assumes that “**said memory**” is a reference to the earlier recitation of “**an alert message cache.**”

2. “Its Memory”

Claim 1 recites “**said client updates the stored value in its memory to reflect the number of additional unread alert messages transmitted from said server to said client.**” *Id.*, 12:67-13:2. Other than the reference to “**said memory**” (noted previously), there is no prior recitation to “**memory**” in claim 1. *Id.*, 12:37-13:8; The specification states:

As the client polls the server it requests the value of the alertID changes.
The client then compares the value of the alertID change with the value stored in *its memory*.

Id., 5:7-11. This Petition assumes that “**its memory**” is a reference to an implied memory of “**said client.**”

3. “Said Processor”

Claim 10 recites “**central server provided with input devices ..., and a database ..., and said processor having program instructions to create and**

store an alert ID value.” *Id.*, 13:37-14:3. Other than the foregoing recitation, the term “**processor**” is not recited in claim 10. *Id.*, 13:34-14:31. The specification and figures make no reference to a processor being part of a particular component. *E.g.*, *id.*, 3:34-36; FIG. 1. This Petition assumes that “**said processor**” is a reference to an implied processor of the “**central server.**”

B. Terms in Claims 10 and 12

Claim 10 and 12 contain arguably indefinite claim terms. To the extent the Board determines at this stage that any claim terms appear indefinite, it should nevertheless institute and require Patent Owner to “provide sufficient explanation or evidence as to why those claims are not indefinite in its patent owner response.”

Vibrant Media, Inc. v. General Elec. Co., IPR2013-00172, Paper 8, at 6-7 (PTAB Jul. 29, 2013). At the institution stage, it is unnecessary for the Board to determine the precise contours of the claim limitation for purposes of this decision. *Amneal Pharm. LLC v. Alkermes Pharma Ireland Ltd.*, IPR2018-00943, Paper 8, at 12-13 (PTAB Nov. 7, 2018).

1. “Get Alert Messages”

Claim 10 recites “**said server requesting a Get Alert Messages using a SOAP communications protocol.**” EX-1001, 14:11-13. While the specification lists other Get commands (including GetUnreadAlerts), the phrase “**Get Alert Messages**” is not recited in the specification. *Id.*, 5:39-54; *see generally id.* And the

'765 Patent does not incorporate any documents (such as documents defining SOAP) by reference. *See generally id.* This Petition assumes that “**Get Alert Messages**” means the GetUnreadAlerts SOAP request that was used in the prior art system. *E.g., id.*, 10:22-23.

2. “Said Client Displays, Display Said Alter Messages”

Claim 12 recites “**said client displays, display said alter messages.**” EX-1001, 14:33-34. The specification does not disclose any “**alter messages**” or the phrase “**displays, display.**” *See generally id.* Moreover, this term was introduced in the original application, and thus, Patent Owner cannot claim a clerical error on behalf of the USPTO. EX-1002 at 319. This Petition assumes that “**said client displays, display said alter messages**” should be construed as said client displays said alert messages.

VII. REQUESTED RELIEF

Petitioner requests institution and cancellation of the Challenged Claims as unpatentable.

VIII. OVERVIEW OF CHALLENGES

A. Statutory Grounds for Challenges

This Petition, supported by the declarations of William Henry Mangione-Smith (EX-1003) and David H. Crocker (EX-1023), is based on the following Grounds:

Ground	Basis for Obviousness Under § 103⁴	Claims
1	<i>Dillon</i> (EX-1005) and AAPA ⁵	1-13
2	<i>Dillon</i> (EX-1005), <i>Brady</i> (EX-1008), and AAPA	1-13
3	<i>Dillon</i> (EX-1005), <i>Douglas</i> (EX-1006), and AAPA	1-13
4	<i>Dillon</i> (EX-1005), <i>Brady</i> (EX-1008), <i>Douglas</i> (EX-1006), and AAPA	1-13

B. Prior-Art Status

Dillon (EX-1005) issued on May 23, 2000, and is prior art under 35 U.S.C. § 102(b).

Douglas (EX-1006) was published on June 25, 2009, and is prior art under 35 U.S.C. §102(b).

⁴ All code citations are pre-AIA.

⁵ USPTO, Updated Guidance on the Treatment of Statements of the Applicant in the Challenged Patent in Inter Partes Reviews Under § 311 (June 9, 2022), available at https://www.uspto.gov/sites/default/files/documents/signed_aapa_guidance_memo.pdf (“AAPA Memo”).

Brady (EX-1008) was filed on September 14, 2009, and published on August 12, 2010. Based on its non-provisional filing date (September 14, 2009), *Brady* is prior art under 35 U.S.C. § 102(e). To the extent Patent Owner attempts to swear behind *Brady*'s non-provisional date, *Brady* claims priority to the *Brady* Provisional (EX-1009), filed on September 12, 2008. As shown in the claim chart below, the *Brady* Provisional provides written description and enablement support for at least claim 1 of *Brady*. EX-1003, ¶¶154-155. Thus, for prior art purposes, *Brady* is entitled to rely on the filing date of the *Brady* Provisional, September 12, 2008. *See Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015).

<i>Brady</i> (EX-1008), Claim 1	Example Support from <i>Brady</i> Provisional (EX-1009)
1. A method of controlling a number of object update requests sent from a client system to an application server, the method comprising the steps of:	¶¶[0010], [0043], [0044], [0045].
receiving a request in a polling server from a client system, the request including data identifying a first user and an object provided to the client system by an application server;	¶¶[0043], [0044], [0045].
determining in the polling server whether an update message related to the object has been received by the polling server from the application server; and if so	¶[0048].
sending a response from the polling server to the client system, the response including information usable by the client system to	¶¶[0045], [0048].

<i>Brady</i> (EX-1008), Claim 1	Example Support from <i>Brady</i> Provisional (EX-1009)
determine if new or updated information related to the object is available on the application server; and	
receiving at the application server an update request from the client system responsive to the client system processing the response to determine that new or updated information related to the object is available on the application server.	¶¶[0045], [0057].

AAPA (discussed below in Section X.A.2.) provides general knowledge of a POSITA, which can be used to supply missing claim limitations, supporting a motivation to combine, or demonstrating the knowledge of a POSITA. AAPA Memo at 6-7.

This Petition also references other documents to demonstrate state of the art, motivation to combine, and/or POSITA's knowledge. EX-1007, EX-1010-EX-1016.

IX. DISCRETIONARY DENIAL IS UNWARRANTED

A. *Fintiv* Weighs Against Denial

Section 314 does not support denial. *First*, this is the first petition Petitioner filed challenging any claim of the '765 Patent. *Cf. Apple Inc. v. Uniloc 2017 LLC*, IPR2020-00854, Paper 9 at 8 (PTAB Oct. 28, 2020) (exercising discretion under §314 and denying institution with respect to follow-on petition where the petitioner had previously filed a petition that the Board declined to institute). *Second*, the

Fintiv factors weigh *against* denial. *Apple, Inc. v. Fintiv, Inc.*, Paper 11, IPR2020-00019 (PTAB Mar. 20, 2020) (precedential).

Factor 1: Neither party has moved for a stay. EX-1019. Absent specific evidence regarding a potential stay, the Board “will not attempt to predict” how a court will proceed. *Sand Revolution II, LLC v. Cont’l Intermodal Group-Trucking LLC*, IPR2019-01393, Paper 24 at 7 (PTAB Jun. 16, 2020) (informative). Factor 1 is therefore *neutral*. *Id.*

Factor 2: The Court has not set a trial date. EX-1019. When no trial date has been set, the Board “will not speculate on when a trial may be scheduled” and found Factor 2 weighs *against* denial. *Tesla, Inc. v. Charge Fusion Techs., LLC*, No. IPR2023-00063, Paper 9 at 32 (PTAB May 11, 2023). The Board may also consider median time to trial.⁶ In the Court, the median time from filing to trial is 39 months for civil cases. EX-1017 at 20. Applying that median time would result in a trial date of around March 2026, which is approximately 8 months *after* the projected statutory deadline for a FWD. Patent Owner caused further delay by filing an amended complaint six months after filing the original. EX-1019 at 4. Factor 2

⁶ USPTO, Memorandum on Interim Procedure for Discretionary Denials in AIA Post-grant Proceedings with Parallel District Court Litigation (June 21, 2022) (“Discretionary Memo”) at 8.

weighs *against* denial. *Unified Patents, LLC v. Justservice.net LLC*, No. IPR2020-01258, Paper 18 at 45-46 (PTAB Feb. 18, 2021) (“in view of the lack of a certain trial date and the proximity between any expected trial date and our final written decision deadline, this factor weighs against denial of institution”).

Factor 3: The “investment in the parallel proceeding by the court and the parties” (*Fintiv*, IPR2020-00019, Paper 11 at 10) weighs against denial. In the Litigation, Petitioner has moved to dismiss and, accordingly, not yet filed an answer.⁷ EX-1019. Because an answer has not been filed, the Court will not issue a scheduling order, and the Parties have not engaged in discovery (including exchange of disclosures and contentions regarding claim construction under the Court’s local rules). EX-1018, Rules 802, 803.1; 804; 805. Accordingly, the Litigation is in its infancy. Factor 3 weighs *against* denial.

Factor 4: If the Board institutes, Petitioner stipulates that it will not pursue an invalidity defense in the Litigation based on the same grounds in this Petition or any grounds that Petitioner could have reasonably raised before the Board in this

⁷ The motion to dismiss (EX-1021) is fully briefed, but the Court has not (i) ruled, (ii) heard oral argument, or (iii) communicated to the parties regarding the motion. EX-1019. Moreover, the motion is directed to ineligibility under 35 U.S.C. § 101, a statute that cannot serve as a basis for this Petition. EX-1021.

Petition. This mitigates concerns of potentially conflicting decisions and duplicative efforts between the district court and the Board. *See Sotera Wireless, Inc. v. Masimo Corp.*, IPR2020-01019, Paper 12 at 18-19 (Dec. 1, 2020) (precedential as to §II.A) “[T]he PTAB will *not* discretionarily deny institution in view of parallel district court litigation where a petitioner presents a stipulation not to pursue in a parallel proceeding the same grounds or any grounds that could have reasonably been raised before the PTAB.” Discretionary Memo at 8. Factor 4 weighs *against* denial.

Factor 5: Petitioner is also the defendant in the Litigation. As noted above, however, the Board is likely to issue a FWD in this proceeding well before the Litigation goes to trial. Accordingly, “estoppel would apply against Petitioner should [the Board] determine that Petitioner failed to demonstrate the unpatentability of the challenged claims” and, in view of that estoppel, Factor 5 weighs *against* denial. *One World Techs., Inc. v. Chervon (HK) Ltd.*, No. IPR2020-00887, Paper 20 at 15 (PTAB Nov. 6, 2020).

Factor 6: The “other circumstances that impact the Board’s exercise of discretion, including the merits” (*Fintiv*, IPR2020-00019, Paper 11 at 6, 14-15) weigh against institution, because the grounds raised are particularly strong: the prior art and AAPA (none of which the Examiner applied) demonstrates unpatentability. Factor 6 weighs *against* denial.

The *Fintiv* factors, viewed holistically, demonstrate that instituting review would promote efficiency and integrity of the system. *Fintiv*, IPR2020-00019, Paper 11 at 6. Discretionary denial under § 314 is *unwarranted*.

B. Section 325(d) Denial is Unwarranted

Section 325 does not apply because Petitioner does not rely on any art or arguments that are the same or substantially the same as those previously presented to the Office. *See Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6 at 8 (PTAB Feb. 13, 2020) (precedential). The three prior art references that form the basis for the Grounds—*Dillon*, *Douglas*, and *Brady*—were not presented or considered during prosecution of the '765 Patent. EX-1001, Cover. The Examiner did not substantively address (or even acknowledge) AAPA during prosecution. *See generally* EX-1002. Accordingly, denial under §325(d) is *unwarranted*.

X. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: *Dillon*-AAPA Renders Obvious Claims 1-13

1. Overview of *Dillon*

Dillon discloses an electronic mail notification system including subscriber terminals for allowing subscribers to retrieve email via an e-mail server. EX-1005, 5:66-6:26. E-mail server includes a subscriber account memory that stores both e-mails and time-stamps associated with each e-mail. *Id.*, 6:4-9, 7:15-29. Time-stamps

include three data types: (i) date/time data of when an e-mail is received (“Date-Time Timestamp”); (ii) data that uniquely identifies an e-mail, such as a version number (“Unique ID Timestamp”); and (iii) a binary indication of whether e-mail has been received for a subscriber (“Binary Timestamp”). *Id.* While *Dillon* describes the three timestamps using the word “[a]lternatively,” *Dillon* does not state that the three timestamps cannot coexist in the same system and/or method. *See id.* Nor does *Dillon* state that the three timestamps are mutually exclusive.

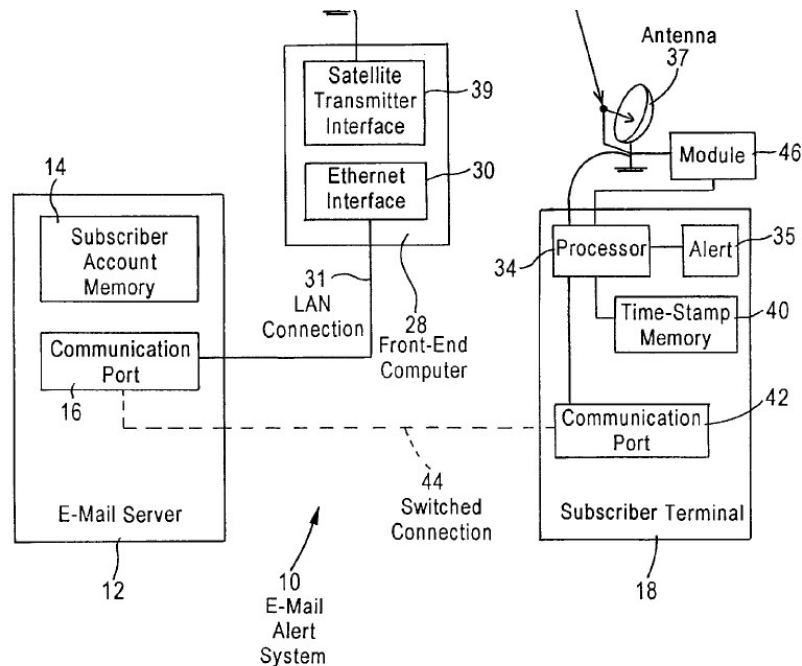


FIG. 2

EX-1005, FIG. 2

On behalf of the e-mail subscribers, front end computer periodically polls the e-mail server and obtains timestamp data. *Id.*, 7:29-41; 5:44-53. While FIG. 2 depicts front end as a separate computer, *Dillon* acknowledges that the front end

may also be functionally equivalent software and that such front-end software may be “remote[]” from the e-mail server. *Id.*, 6:27-32. *Dillon* discloses a comparison of the time-stamp received from the e-mail server with the “latest” timestamp stored in the time-stamp memory. *Id.*, 8:66-9:12. The subscriber terminal can be programmed to retrieve e-mail messages when the comparison between the timestamp values differs. *Id.*, 9:21-27.

2. Overview of AAPA

AAPA consists of the following.

First, the specification admits that Patent Owner’s “Desktop Alert system” is “*prior art*.” EX-1001, 9:39-40 (referencing “*prior art* Desktop Alert system”); *see also id.*, 2:45-47 (describing FIG. 3 as “depicting the processor activity on a *prior art* system before the improvements have been applied”); 6:11-24 (describing FIGS. 2 and 3 as “Performance in a *Prior Art* System”); *accord* EX-1022 at 8-9 (labeling FIGS. 3 and 10 as “Prior Art” and referring to Desktop Alert system as “*prior art* system”); 10 (referencing the “*prior art* Desktop Alert web application/service+SQL Server”). The specification distinguishes between (a) the prior art Desktop Alert system as an example of a “legacy system[] that uses conventional polling architecture” and (b) improvements to the purported invention. EX-1001, 1:10-17.

Second, the specification admits conventional polling models (the prior art Desktop Alert system being one example) were “*widely adopted*” in the prior art and particularly that polling models were “*commonly used* for checking email and other applications that require the timely updating of information such as stock prices and weather.” *Id.*, 3:14; 4:18-20.

Third, the specification admits that polling at predetermined frequency was known in the prior art. *Id.*, 3:11-14: The specification admits that “[t]he *basic concept* behind [prior art] polling models is that the client *periodically* provides *queries* to the host or server requesting updated messaging information.” *Id.*, 3:8-17. The specification also admits “polling frequency” can be “determined in *existing* polling based systems” and provides an exemplary polling frequency of “once per minute.” *Id.*, 4:20-23. The specification further admits that in existing polling-based systems, the “*polling frequency can be set* by knowing the number of expected clients and the available performance (potential requests/second) of the selected hardware.” *Id.*, 4:23-26; *see also id.*, 4:42-49 (describing that in the conventional polling system the GetUnreadAlerts is “the call initiated by the client to the server and is communicated to the server *at the polling frequency*”).

Fourth, the specification admits that the GetUnreadAlerts SOAP request was used in the prior art Desktop Alert system and/or the “conventional polling system.” *Id.*, 4:39-49; *see also id.*, 4:41-42 (“the Desktop Alert web service

GetUnreadAlerts call”); 4:46-48 (“critical GetUnreadAlerts call ... in the *conventional* polling system”); 4:50-51 (“[i]n the Desktop Alert system, each call to GetUnreadAlerts results in”); 5:59-61 (referencing the number of calls using the “*Existing* GetUnreadAlerts”); 6:7 (Table 1) (“*Existing* GetUnreadAlerts”); 6:11-24 (describing calls to GetUnreadAlerts in the “*Prior Art* System”); 10:22-23 (“the client starts with the GetUnreadAlerts SOAP request as has been used in the *prior art* systems”).

Fifth, the specification admits that the prior art Desktop Alert system “shares communications and commands using *standardized* protocols” and that such *standardized* protocols include “XML and HTML.” *Id.*, 1:17-19; *accord id.*, 1:14-17 (stating that the prior art Desktop Alert system uses “*IP-Based alert* technology that can be integrated with enterprise-class environments”).

Sixth, in the Litigation, Patent Owner has admitted that various protocols (including SOAP, HTTP, HTTP GET, RPC, RPC/UDP, and RPC/TCP) recited in the claims were “*known* computer communication protocols” and admitted that SOAP is a “prior art conventional polling protocol.” EX-1022 at 15, 23. Patent Owner made similar admissions in the ’765 Patent. EX-1001, 4:44 (stating that GetUnreadAlerts (which is used the prior art Desktop Alert system as noted above) “is a SOAP call”); 8:5-6-7 (“SOAP is an XML-based messaging protocol”); 10:22-23 (“the client starts with the GetUnreadAlerts *SOAP* request as has been used *in*

the prior art systems”); 11:20-22 (referencing the “*prior art* SOAP system”); *see also* EX-1002 at 318 (filed application including claim that recites “*Conventional* SOAP protocol”).

Seventh, the specification admits that the prior art Desktop Alert system made calls to a database. *Id.*, 4:46-49. The specification further admits that, at the time of the invention, “*industry standard* database[s]” existed including SQL™, MySQL™, and Oracle™ databases. *Id.*, 3:44-45; FIG. 12 (referencing “SQL Statement use, *before*”); 2:66-67 (describing FIG. 12 as “depicting SQL statement use, *before* the improvements”).

3. Reasons to Combine *Dillon* and AAPA⁸

POSITA would have been motivated to combine *Dillon* and AAPA.

First, *Dillon* and AAPA are analogous prior art to the '765 Patent and are in the same field of *polling* systems. EX-1003, ¶43. The '765 Patent is “directed to manners in which to improve the performance of legacy systems [*i.e.*, including

⁸ The AAPA Memo states that AAPA may be used for “supplying missing claim limitations that were generally known in the art” and does not impose a requirement of showing a reason or motivation to combine AAPA with prior art. AAPA Memo at 6. Petitioner nevertheless provides a rationale if the Patent Owner argues otherwise.

the AAPA] that use a conventional *polling* architecture.” EX-1001, 1:10-12.

Dillon is directed to a “*polling* means for *polling* the e-mail server...to determine whether e-mail for an e-mail alert service subscriber has been received by the e-mail server.” EX-1005, 6:35-38.

Second, a POSITA would have been motivated to utilize conventional aspects (such as use of known / standardized protocols and industry standard databases) taught by the AAPA in the implementation of network details left unspecified by *Dillon*. EX-1003, ¶44. *Dillon* discloses an e-mail server, subscriber terminals, and front-end software with a polling means that periodically polls the e-mail server and obtains time-stamps (which can be the Unique ID Timestamp). EX-1005, 5:66-6:26; 7:29-41; 5:44-53. *Dillon* discloses comparing the most-recently-received Unique ID Timestamp with the latest-stored Unique ID Timestamp in memory. *Id.*, 8:66-9:12. When the two Unique ID Timestamp values are found to differ, the e-mail server sends the subscriber terminal e-mails. *Id.*, 8:66-9:12, 9:21-27. But *Dillon* does not provide specific implementation details (such as the specific use of HTTP packets or whether communication between the subscriber terminal and e-mail server use the SOAP, RPC, RPC/UDP, RPC/TCP protocols) for the form of the communications between the various network components. These protocols were well-known and HTTP in particular was “almost universally used on the Web.” EX-1022 at 15; EX-1016 at 259, 449, 487-88, 513, 536. A POSITA therefore would have been motivated

to utilize the conventional networking implementations of the AAPA. Specifically, a POSITA would have been motivated to combine *Dillon* with the following aspects of the AAPA:

- GetUnread Alerts SOAP request was used in the prior art Desktop Alert system and/or the “*conventional* polling system.” EX-1001, 4:39-49; 4:41-42; 4:46-48; 4:50-51; 5:59-61; 6:7 (Table 1); 6:11-24; 10:22-23;
- sharing “communications and commands using *standardized* protocols” and that such standardized protocols include “XML and HTML.” *Id.*, 1:17-19;
- the prior art Desktop Alert system making calls to a database and the admission of the existence in the prior art of “*industry standard* database[s]” including SQL™, MySQL™, and Oracle™ databases. *Id.*, 3:44-45; FIG. 12; 2:66-67; 4:46-49; and
- employing “*known* computer communication protocols” including SOAP, HTTP, HTTP GET, RPC, RPC/UDP, and RPC/TCP recited in the claims.” EX-1022 at 15, 23; *see also* EX-1001, 4:44; 8:5-6-7; 10:22-23; 11:20-22; *see also* EX-1002 at 318.

These implementation details and/or modifications would have involved combining prior art elements (*i.e.*, *Dillon*’s polling system) according to known methods (*i.e.*,

the implementation details admitted as prior art by the AAPA) to yield the predictable result (*i.e.*, polling for and retrieval of retrieving unread e-mails with reduced database usage and increased polling frequency). *See KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 416 (2007).

Third, a POSITA would have been motivated to combine *Dillon*’s polling system by incorporating AAPA’s implementation details given the AAPA’s admissions that the implementations were “known,” “conventional,” “standardized,” and “industry standard[],” and “well-known.” EX-1003, ¶45 (citing EX-1001, 1:17-19; 2:66-67; 3:44-45; 4:39-49; 4:41-42; 4:46-49; 4:50-51; 5:59-61; 6:7 (Table 1); 6:11-24; 10:22-23; FIG. 12). A POSITA would have understood that these and other communication standards constituted a finite number of predictable solutions, and a POSITA would have been motivated to pursue these known options within a POSITA’s technical grasp. *KSR*, 550 U.S. at 421.

Fourth, a POSITA would have had reasonable expectations of success given that *Dillon* and AAPA build upon conventional and standardized networking technologies and protocols employed for their intended purposes. EX-1003, ¶46. For instance, a POSITA would have little doubt that communicating by these known protocols (which is taught by the AAPA) would be a reliable and efficient form given their widely-used and pervasive nature. A POSITA would thus have had a reasonable expectation of success in combining *Dillon* with

AAPA. *Regents of Univ. of California v. Broad Inst., Inc.*, 903 F.3d 1286, 1291 (Fed. Cir. 2018).

4. Claim 1

- a) **[1.1] An improved method for the dissemination of an alert message to a plurality of personal computers using a network communications protocol**

Dillon-AAPA renders obvious [1.1]. EX-1003, ¶¶47-51.

First, Dillon discloses or suggests a “**plurality of personal computers.**”

Dillon discloses a “subscriber terminal 18 may ... be a stand-alone personal computer” and references “subscriber terminals” (plural). EX-1005, 6:17-18; 9:65; FIG 2.

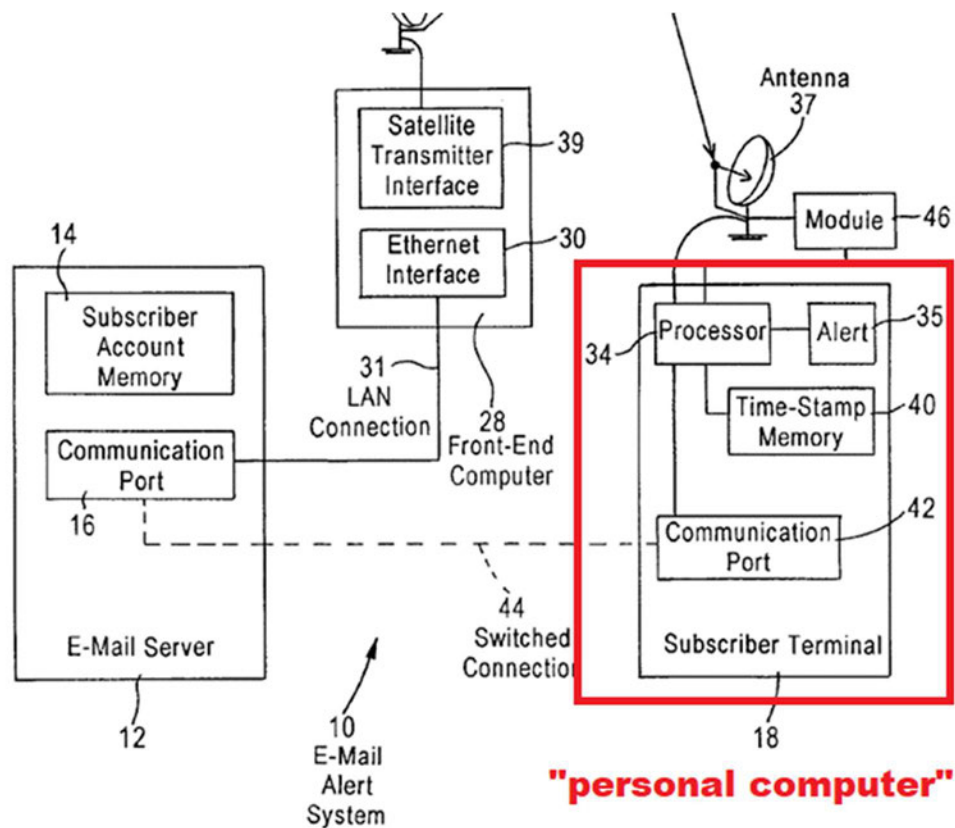


FIG. 2

EX-1005, FIG. 2

Second, Dillon discloses or suggests “**dissemination of an alert message...using a network communications protocol.**” *Dillon* describes a “subscriber terminal 18 includes a communication port 42...for retrieving e-mail from the e-mail server 12...e-mail may be downloaded via a link 44 [that] is a terrestrial connection, such as...a packet-based networking connection.” *Id.*, 9:39-50. The e-mail server “administers e-mail accounts, such as internet-accessible post-office-protocol 3 (POP-3) e-mail accounts.” *Id.*, 7:1-4. To the extent that *Dillon* does not disclose or suggest “**alert message,**” a POSITA would have been motivated to

modify the emails of *Dillon* to constitute alert messages. AAPA acknowledges that in the prior art Desktop Alert system “[a]lerts” can be sent as emails. EX-1001, 1:50-51. Using e-mails for alert messages was well known in the art. EX-1013, 3:14-28 (broadcasting emergency alerts in “possible” forms including “emails”); EX-1015, ¶[0009] (“alert message is displayed as...an email”).

Third, to the extent *Dillon* does not disclose or suggest dissemination “**using a network communications protocol**,” AAPA admits that the prior art Desktop Alert system “uses an *IP-Based* alert technology ... shares communications and commands using standardized protocols such as XML and HTML,” and “allows transmission of such [alert] messages to a plurality of clients.” EX-1001, 1:14-25. The AAPA also admits that SOAP is a conventional protocol in the file history. EX-1002 at 318 (filed application including claim that recites “Conventional SOAP protocol”). Finally, the AAPA admits that various protocols (including SOAP, HTTP, HTTP GET, RPC, RPC/UDP, and RPC/TCP) recited in the claims were “*known* computer communication protocols.” EX-1022 at 15.

Therefore, *Dillon*-AAPA renders obvious dissemination of an alert message (*i.e.*, e-mail of *Dillon* and/or an alert message of the AAPA) to a plurality of personal computers (*i.e.*, subscriber terminals) using a network communications protocol (*i.e.*, *Dillon*’s IP-based protocol, packet-based protocol, or POP-3 protocol and/or the AAPA’s standardized protocols, such as XML and HTML).

- b) [1.2] said computers comprising clients, each said clients subscribing to a central server, said server comprising an alert message cache, said server adapted to receive and store alert messages, and

Dillon-AAPA renders obvious [1.2]. EX-1003, ¶¶52-59.

First, Dillon discloses or suggests “**said computers comprising clients, each said clients subscribing to a central server.**” As depicted in FIG. 2, *Dillon* discloses an “e-mail server having a subscriber account memory and being accessible by the subscriber via a subscriber terminal.” EX-1005, 5:33-34. *Dillon* also discloses “e-mail server 12 administers e-mail accounts...for a plurality of subscribers to an e-mail service.” *Id.*, 7:1-4.

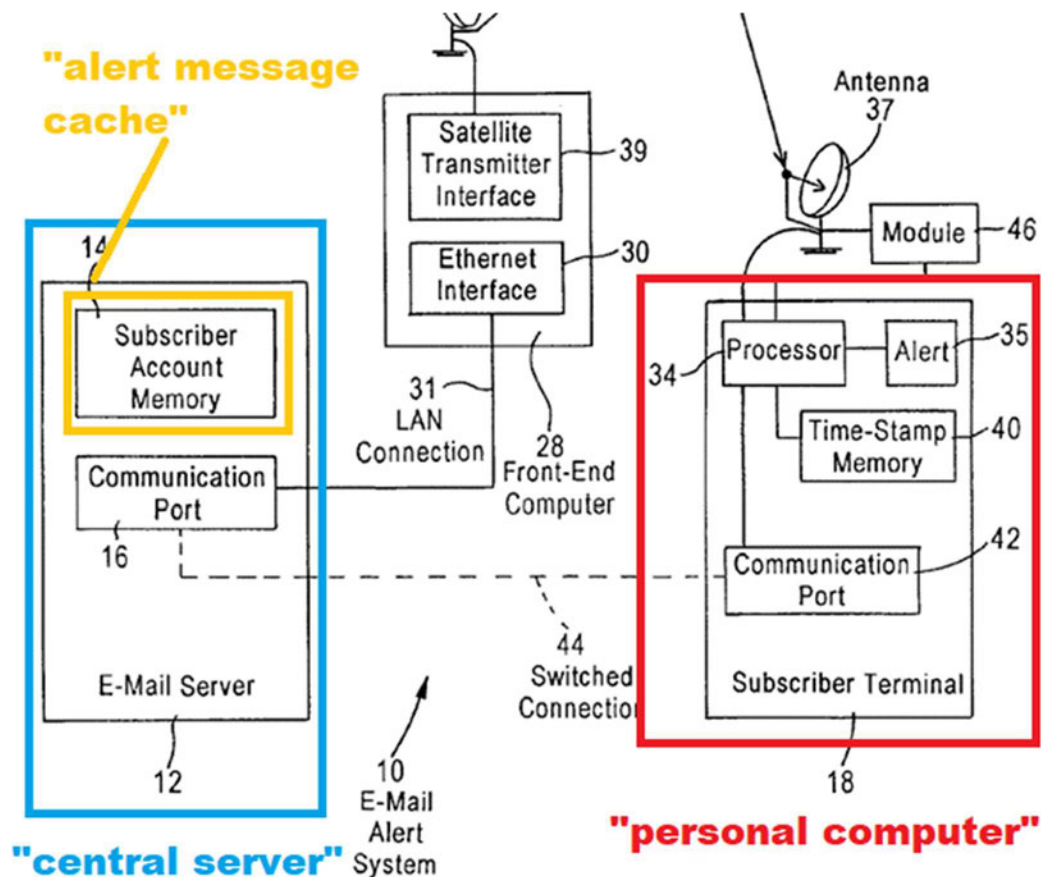


FIG. 2

EX-1005, FIG. 2

Second, Dillon discloses or suggests “said server comprising an alert message cache, said server adapted to receive and store alert messages.” As depicted in FIG. 2, Dillon describes that the “e-mail server includes a subscriber account memory” which may “be used to store e-mail messages for e-mail service subscribers.” *Id.*, 3:26-35. Dillon describes that “when the e-mail server 12 receives e-mail for an e-mail service subscriber, the subscriber account memory 14....stor[es] the e-mail message itself.” *Id.*, 6:4-9.

Thus, *Dillon*-AAPA renders obvious computers comprising clients (*i.e.*, subscriber terminals), each said clients subscribing to a central server (*i.e.*, e-mail server), said server comprising an alert message cache (*i.e.*, subscriber account memory), said server adapted to receive and store alert messages (*i.e.*, e-mails).

c) **[1.3] and said server adapted to generate and store alert ID numbers,**

Dillon-AAPA renders obvious [1.3]. EX-1003, ¶¶56-59.

First, *Dillon* discloses or suggests “**alert ID numbers.**” *Dillon* describes a “time-stamp [that] may...be assigned a value that uniquely identifies a particular e-mail message received by the e-mail server for an e-mail service subscriber.... it is preferable that the time-stamp be assigned a value that changes each time an e-mail message is received and which is not likely to repeat as multiple e-mail messages are received.” EX-1005, 4:28-36; 7:23-26 (“a time-stamp may be...a version number, that uniquely identifies a particular e-mail message received by the e-mail server 12 for the given e-mail alert service subscriber.”). This Petition refers to the foregoing disclosure in *Dillon* as the “Unique ID Timestamp.”

Second, *Dillon* discloses or suggests that the “**said server adapted to generate and store**” the alert ID numbers (*i.e.*, Unique ID Timestamp). *Dillon* describes an “e-mail server...updates the time-stamp memory...for an e-mail service subscriber each time the e-mail server...receives e-mail for the e-mail service

subscriber.” *Id.*, 8:34-39. The “subscriber account memory...of the e-mail server...contains [the] time-stamp.” *Id.*, 7:15-16.

Therefore, *Dillon*-AAPA renders obvious a server (*i.e.*, e-mail server) that is adapted to generate and store alert ID numbers (*i.e.*, Unique ID Timestamp).

- d) **[1.4] said method comprising a first triggering step comprises the use of a HTTP packet comprising a request for a Boolean value relating to whether there are new alert messages and said packet comprises approximately 20 bytes**

Dillon-AAPA renders obvious [1.4]. EX-1003, ¶¶60-65.

First, *Dillon* discloses or suggests a “**request for a Boolean value relating to whether there are new alert messages.**” *Dillon* describes “[w]hen the front-end computer 28 polls the e-mail server 12 on behalf of one or more e-mail alert service subscribers as described above, it obtains from the e-mail server 12, for each e-mail alert service subscriber, an indication, such as the time-stamp stored in the subscriber account memory 14, of whether e-mail has been received by the e-mail server 12.” EX-1005, 7:30-41. *Dillon* describes that “the time-stamp could...be a repeatable value, such as a binary (*e.g.*, one or zero) indication of whether or not e-mail for an e-mail service subscriber has been received by the e-mail server.” *Id.*, 4:36-41. This Petition uses the term “Binary Timestamp” to refer to the foregoing disclosure. A POSITA would understand the *Dillon*’s reference to time-stamp being a “binary (*e.g.*, one or zero)” to disclose or suggest a “**Boolean value.**” A POSITA

would understand that computers “use integer data types to implement Boolean values, usually (although not always) with 0 equaling false and “not 0” equaling true.” EX-1016 at 67. A POSITA would also understand that the two digits, 0 and 1, in a binary number system “can represent the logical values true and false...” *Id.* at 57.

Second, to the extent that Patent Owner argues that the Unique ID Timestamp and Binary Timestamp are listed as “alternative[s]” to each other and thus should not coexist, nothing in *Dillon* describes the two alternatives as mutually exclusive. EX-1005, 7:15-29. A POSITA would understand *Dillon* as disclosing the use of both time-stamps to coexist in the same method or system. Even if both time-stamps were mutually exclusive in *Dillon*, a POSITA would be motivated to modify *Dillon* to coexist in the same method of system. For instance, the Binary Timestamp reduces messaging bandwidth and other resources which was a well-recognized problem. EX-1014, ¶¶[0006], [0009] (“[w]hat is needed is a solution that reduces the amount of bandwidth that is required to retrieve new e-mail from a mail server”); EX-1007, ¶[0022] (“Sending simple notifications that do not include a large amount of data reduces the overhead imposed on object server”). But the Binary Timestamp does not have the Unique ID Timestamp’s ability of to “uniquely identif[y] a particular e-mail message received by the e-mail server” of the uniquely identifiable time stamp. EX-1005, 4:28-36.

Third, AAPA discloses or suggests “**triggering step comprises the use of a HTTP packet comprising a request.**” AAPA admits that “**HTTP**” was a “*known* computer communication protocol[.]” EX-1022 at 15. A POSITA would understand that AAPA’s admission of HTTP as known protocol would also include using an HTTP packet comprising a request. EX-1008, ¶[0028] (describing an “HTTP client...for sending and receiving HTTP messages to and from an HTTP server” and specifically discloses “incoming HTTP requests”). Such “requests” are sent by a “polling client ... that ... calls or queries the polling server”. *Id.*, ¶¶[0053]-[0057]. A POSITA would also understand that such an HTTP packet comprising a request could constitute a “**triggering**” step because the use of the HTTP packet comprising a request can result in a function or process being initiated. *Id.*, ¶[0055] (“If the id is new, the polling client should cause the page to be refreshed.”).

Fourth, to the extent that *Dillon-AAPA* does not disclose or suggest “**packet comprises approximately 20 bytes,**” it would have been obvious to POSITA to modify *Dillon-AAPA* for such a packet size. HTTP structure was well-understood in the art. Based on the HTTP standards, POSITA would understand that the shortest possible HTTP request is a simple GET method, made by connecting directly to a specific server. EX-1010, §8.1; EX-1011, §9.3; *see also* EX-1023. For HTTP 1.0, the presence of headers is expected, so to signify the end of the headers you need an empty line. The shortest request is then:

```
GET / HTTP/1.0<CR><LF>
<CR><LF>
```

which results in a total of 18 bytes. For HTTP 1.1, POSITA would understand the need for a Host header. POSITA would understand that the minimum initial request line is:

```
GET / HTTP/1.1<CR><LF>
```

which is 16 bytes (including the two invisible CRLF bytes). A minimum Host line is:

```
Host : x<CR><LF>
```

The above consists of a one-byte hostname which results in 8 bytes (again two CRLF bytes). To signify the end of headers, it is necessary for another CRLF, which would consist of another 2 bytes. This would result in a minimum HTTP request size of 26 bytes (16 bytes+8 bytes+2 bytes). Accordingly, POSITA would have understood that the minimum size of an HTTP request is in the range of 18 to 26 bytes, which is in the range of “**said packet comprises approximately 20 bytes.**”

The “**packet comprises approximately 20 bytes**” would have been obvious because POSITA would have been motivated to reduce packet size. EX-1014, ¶¶[0006], [0009] (noting that “frequent polling increases the traffic capacity requirement of the network, as well as the computational resource usage” and concluding that “[w]hat is needed is a solution that reduces the amount of bandwidth that is required to retrieve new e-mail from a mail server”); *Id.*, ¶[0028] (noting that

“short” command “reduces data communications access overhead and reduces delivery latency”); EX-1007, ¶¶0022] (“Sending simple notifications that do not include a large amount of data reduces the overhead imposed on object server”). The “discovery of an optimum value of a variable in a known process is usually obvious.” *Pfizer, Inc. v. Apotex, Inc.*, 480 F.3d 1348, 1368 (Fed. Cir. 2007). The experimentation needed, then, to arrive at the claimed “**20 bytes**” was “nothing more than routine” application of a well-known problem-solving strategy, “the work of a skilled [artisan], not of an inventor.” See *DyStar Textilfarben GmbH v. C.H. Patrick Co.*, 464 F.3d 1356, 1371 (Fed.Cir.2006).

Therefore, *Dillon*-AAPA renders obvious the use of a HTTP packet comprising a request for a Boolean value (*i.e.*, Binary Timestamp) relating to whether there are new alert messages and said packet comprises approximately 20 bytes.

- e) **[1.5] a second step wherein said client contacts the central server and requests the value of said alert ID number in said memory, said value reflecting the number of alert messages received by said central server**

Dillon-AAPA renders obvious [1.5]. EX-1003, ¶¶66-70.

First, Dillon discloses or suggests “**said client contacts the central server and requests the value of said alert ID number in said memory.**” *Dillon* discloses that the “*the time-stamp stored in the subscriber account memory* of the e-

mail server.” EX-1005, 7:33. *Dillon* discloses that “the polling means of the front-end *polls the e-mail server periodically to obtain the time-stamp for the e-mail service subscriber.*” *Id.*, 13:8-9.

Second, while *Dillon* depicts the front-end computer separate from the subscriber terminal in Figure 2 (reproduced above), POSITA would have been motivated to modify *Dillon* to install computer program equivalent of the front end on the subscriber terminal. *Dillon* specifically states that the term “front-end” includes software implementations:

Therefore, when used by itself herein, the term “front-end” generically encompasses not only front-end computers (*i.e.*., hardware together with any associated software) but also equivalent software-only implementations of front-end computer functionality.

EX-1005, 3:20-25; *see also id.*, 3:36-37 (“e-mail alert front-end computer, or a computer program equivalent thereto”). Moreover, *Dillon* discloses that both the front-end computer and subscriber terminal may be “remote” from the e-mail server. *Id.*, 6:30-32; 5:34-35. POSITA would be motivated to install *Dillon*’s computer program equivalent to the front-end computer on *Dillon*’s subscriber terminal (*i.e.*., personal computer) to simplify the system design, reduce the number of hardware components, which would be less costly and more efficient.

Third, *Dillon* discloses that “a time-stamp may be...a version number, that uniquely identifies a particular e-mail message received by the e-mail server 12 for

the given e-mail alert service subscriber.” *Id.*, 7:23-26. *Dillon* also states that the “[p]referably, the time-stamp stored in the subscriber account memory of the e-mail server is updated each time e-mail for the e-mail service subscriber is received.” *Id.*, 4:49-54. POSITA would understand the foregoing references to indicate that the time-stamp is incremental in nature and thus indicates **“said value reflecting the number of alert messages received by said central server.”**

Therefore, *Dillon*-AAPA renders obvious client (*i.e.*, the subscriber terminal with the computer program equivalent of the front end installed) contacting the central server (*i.e.*, e-mail server) and requests (*i.e.*, polls) the value of said alert ID number (*i.e.*, Unique ID Timestamp) in said memory (*i.e.*, subscriber account memory), said value reflecting the number of alert messages received by said central server (*i.e.*, Unique ID Timestamp).

f) [1.6] a third step wherein said server transmits said value to said client

Dillon-AAPA renders obvious [1.6]. EX-1003, ¶¶71-73.

Dillon discloses that “the polling means of the front-end polls the e-mail server periodically to obtain the time-stamp for the e-mail service subscriber.” EX-1005, 4:49-54. *see also id.*, 9:7-12 (referencing the comparison of “the time-stamp received from the e-mail server” with other values at the subscriber terminal).

Therefore, *Dillon-AAPA* renders obvious that the server (*i.e.*, e-mail server) transmits (*i.e.*, obtains) the value (*i.e.*, value of the Unique ID Timestamp) to the client (*i.e.*, the subscriber terminal).

- g) [1.7] a fourth step wherein said client compares a stored value at the client with the value of the transmitted alert ID number, and**

Dillon-AAPA renders obvious [1.7]. EX-1003, ¶¶74-46.

Dillon explains that “the subscriber terminal 18 determines....whether e-mail for a subscriber has been received by the e-mail server...the subscriber terminal 18 includes a time-stamp memory 40 for storing a ‘latest’ time-stamp which the subscriber terminal 18 most recently received in an e-mail notification from the front-end computer 28 for the e-mail alert service subscriber. The *comparison of the time-stamp received from the e-mail server 12 with the most recently received or ‘latest’ time-stamp stored in the time-stamp memory 40 of the subscriber terminal 18* is then made by the processor 34 or other suitable comparing means provided at the subscriber terminal 18.” EX-1005, 8:66-9:12.

Therefore, *Dillon-AAPA* renders obvious the client (*i.e.*, the subscriber terminal) comparing a stored value at the client (*i.e.*, “latest” time-stamp stored in the time stamp memory of the subscriber terminal) with the value of the transmitted alert ID number (*i.e.*, the time-stamp received from the e-mail server).

- h) **[1.8] if said transmitted value exceeds said stored value, a fifth step wherein said client transmits a second message to said server and requests for the server to transmit all unread alert messages in a GetUnreadAlerts request from said server that are specific to said client, and**

Dillon-AAPA renders obvious [1.8]. EX-1003, ¶¶77-80.

First, Dillon discloses or suggests “**if said transmitted value exceeds said stored value, a fifth step wherein said client transmits a second message to said server and requests for the server to transmit all unread alert messages ... from said server that are specific to said client.**” *Dillon* discloses that the subscriber terminal “can be suitably programmed to automatically dial in to the e-mail server 12 and retrieve e-mail messages for an e-mail alert service subscriber whenever the subscriber terminal 18 detects that the received time-stamp differs from the stored latest time-stamp.” EX-1005, 9:13-27. While *Dillon* uses the word “differs” (but not “**exceeds**”), given the sequential nature of the value of the Unique ID Timestamp, POSITA would have understood that “differs” would disclose or suggest “**exceeds.**” Moreover, to the extent that *Dillon* does not disclose or suggest that the subscriber terminal “**transmits a second message to said server,**” POSITA would be motivated to modify *Dillon*’s to also encompass “**exceeds**” given that a difference is either a higher (“**exceeds**”) or lower value. While *Dillon* does not state that “**all unread**” messages are retrieved, POSITA would understand *Dillon* as suggesting that all unread messages are sent because any difference between the received time

stamp and the stored latest time stamp will mean that new messages have been received and it would be more efficient to retrieve all unread messages in one communication. Moreover, a POSITA would have understood *Dillon*'s reference to "retrieve e-mail messages" to include "**all unread**" messages.

Second, while *Dillon* does not explicitly reference "**GetUnreadAlerts request**," the AAPA admits that the GetUnreadAlerts request was used in the prior art Desktop Alert system and/or the "conventional polling system." *Id.*, 4:39-49; *see also id.*, 4:41-42 ("the Desktop Alert web service GetUnreadAlerts call"); 4:46-48 ("critical GetUnreadAlerts call ... in the conventional polling system"); 4:50-51 ("[i]n the Desktop Alert system, each call to GetUnreadAlerts results in"); 5:59-61 (referencing the number of calls using the "Existing GetUnreadAlerts"); 6:7 (Table 1) ("Existing GetUnreadAlerts"); 6:11-24 (describing calls to GetUnreadAlerts in the "Prior Art System"). To the extent that AAPA did not disclose or suggest "**GetUnreadAlerts**," Get Attribute operations (which would encompass GetUnreadAlerts) were well known in the art. EX-1007, ¶[0024] (describing "Get Attribute operations" that "can provide information such as attributes that were changed").

Therefore, *Dillon*-AAPA renders obvious that if said transmitted value (*i.e.*, received time-stamp) exceeds (*i.e.*, differs, which would have been understood by POSITA as including exceeds) said stored value (*i.e.*, latest time-stamp stored in the

time-stamp memory), said client (*i.e.*, subscriber terminal) transmits a second message to said server and requests for the server to transmit all unread alert messages in a GetUnreadAlerts request from said server that are specific to said client.

- i) **[1.9] if there are unread alert messages specific to said client in said server, a sixth step wherein said server sends to client all unread alert messages specific to the client, and**

Dillon-AAPA renders obvious [1.9]. EX-1003, ¶¶81-84.

First, Dillon discloses “the subscriber terminal 18 optionally can be suitably programmed to automatically dial in to the e-mail server 12 and retrieve e-mail messages for an e-mail alert service subscriber whenever the subscriber terminal 18 detects that the received time-stamp differs from the stored latest time stamp.” EX-1005, 9:13-27.

Second, while Dillon does not reference sending “**all unread messages**,” for the same reasons discussed with respect to [1.8]’s recitation of “**all unread messages**,” POSITA would understand that all unread messages are being sent by *Dillon*’s e-mail server. *See* Section X.A.4.h.

Therefore, *Dillon-AAPA* renders obvious that if there are unread alert messages specific to said client in said server, the server (*i.e.*, e-mail server) sends all unread alert messages to the client (*i.e.*, e-mail service subscriber) when there are unread alert messages specific to the said client.

- j) **[1.10] a further step wherein said client updates the stored value in its memory to reflect the number of additional unread alert messaged transmitted from said server to said client in response to said second message, and**

Dillon-AAPA renders obvious [1.10]. EX-1003, ¶¶85-88.

First, Dillon discloses or suggests “**said client updates the stored value in its memory.**” *Dillon* discloses “the subscriber terminal 18 includes a time-stamp memory 40 for storing a ‘latest’ time-stamp which the subscriber terminal 18 most recently received in an e-mail notification from the front-end computer 28 for the e-mail alert service subscriber.” EX-1005, 8:66-9:7.

Second, Dillon discloses or suggests that the value stored in memory “**reflect[s] the number of additional unread alert messaged transmitted from said server to said client in response to said second message.**” POSITA would understand that the number of additional unread messages could be determined with the sequential Unique ID Timestamp of *Dillon*. For instance, if the “latest” stored time stamp is 1 (representing the 1st message) and the received time stamp is 5 (representing the 5th message), POSITA would understand that the number of unread alert messages could be calculated from the difference of both numbers (4). Thus, the value stored in memory, “**reflect[s] the number of additional unread alert messages transmitted from said server to said client in response to said second message.**”

Therefore, *Dillon*-AAPA renders obvious a client (*i.e.*, subscriber terminal) that updates the stored value (*i.e.*, stores the “latest” Unique ID Timestamp) in its memory (*i.e.*, time-stamp memory) to reflect the number of additional unread alert messages transmitted from said server to said client in response to said second message.

- k) **[1.11] wherein, upon the reception of said unread alert messages in said sixth step, said alert messages are in instant message format and said client opens a window on a display panel to display said alert messages,**

Dillon-AAPA renders obvious [1.11]. EX-1003, ¶¶89-93.

First, *Dillon* discloses or suggests “**reception of said unread alert messages**” for the earlier reasons discussed with respect to transmitting “**unread alert messages**” to the client (*i.e.*, the client receives the transmission). *See* Section X.A.4.h.

Second, the AAPA discloses or suggests “**said client opens a window on a display panel to display said alert messages.**” Specifically, the AAPA admits that the Desktop Alert system had the “feature[] and capability” to “[s]end alerts that open with ‘upper most screen’ functionality. If a user has multiple applications open, the alert will take the fore screen and stay open until engaged.” EX-1001, 1:24-63. The AAPA also admits that the prior art Desktop Alert system is a “Windows client” that can “[s]end Alerts to Desktops.” *Id.*, 1:44-50. While the AAPA is silent as to

opening a window, POSITA would understand the AAPA's reference to "open with 'upper most screen'" in the context of a Windows client as disclosing or suggesting **"said client opens a window on a display panel to display said alert messages."**

Third, Dillon discloses or suggests **"instant message format."** Specifically, *Dillon* discloses that "[i]f the received time-stamp differs from the latest time-stamp stored in the time-stamp memory 40, then the alert 35 provides the e-mail alert service subscriber with an audible and/or visual indication that e-mail for the e-mail alert service subscriber has been received by the e-mail server 12. For example, a blinking icon may appear on a display screen (not shown) of the subscriber terminal 18 until a user acknowledges having seen the e-mail notification (*e.g.*, by clicking the blinking icon with a mouse (not shown))." EX-1005, 9:13-27. POSITA would have understood this reference in *Dillon* to encompass **"instant message format"** because the audible and/or visual indication (*e.g.*, blinking icon) appears on the screen without opening (*i.e.*, instantly). Moreover, to the extent *Dillon* does not disclose or suggest "instant message format," it would have been obvious to POSITA to substitute the e-mail of *Dillon* with instant messages given that both instant messages and emails were known to be interchangeable in the context of alert messages. EX-1015, claim 25 (claiming an "alert message is an email message, instant message, or mobile message"). Given the AAPA's disclosure to open a window to display an alert message in that "open with 'upper most screen'

functionality,” POSITA would have found it obvious to format the messages in a well-known and commonly used “**instant message format**,” which would have been a simple modification.

Therefore, *Dillon*-AAPA renders obvious that upon the reception of said unread alert messages in said sixth step, said alert messages (*i.e.*, e-mail messages) are in instant message format and said client (*i.e.*, subscriber terminal) opens a window on a display panel to display said alert messages.

I) [1.12] wherein said communications protocol comprises protocols for the transmission of messages over the internet.

Dillon-AAPA renders obvious [1.12]. EX-1003, ¶¶94-97.

First, *Dillon* discloses or suggests “**transmissions of messages over the internet.**” In particular, *Dillon* discloses that the that e-mails have “proliferated in the past several years ... in the environment of wide-area networks such as the internet.” EX-1005, 1:24-28. The “e-mail server....administers e-mail accounts, such as internet-accessible post-office-protocol 3 (POP-3) e-mail accounts.” *Id.*, 7:1-3. The link 44 interconnects “communication port 16 of the e-mail server 12 with the communication port 42 of the subscriber terminal 18.” *Id.*, 9:42-45. *Dillon* discloses that link 44 “a packet-based networking connection” and one of its dependent claims recites that “wherein the packet-switched network is the Internet.” *Id.*, 9:45-51, 15:37-38.

Second, for the same reasons discussed with respect to [1.2], *Dillon-AAPA* renders obvious “**communications protocol**” for transmission of messages over the Internet. *See* Section X.A.4.a.

Therefore, *Dillon-AAPA* renders obvious the use of communication protocols for transmission of messages over the internet.

5. Claim 2

[2.0] The method recited in claim 1 wherein said client will contact said server at predetermined polling frequencies.

Dillon-AAPA renders obvious [2.0]. EX-1003, ¶¶98-101.

First, *Dillon* discloses “[t]he e-mail alert front-end computer....includes polling means....for polling the server to determine whether e-mail for the service subscriber has been received by the server....[p]referably, the polling means polls the server *periodically*.” EX-1005, 3:36-46. POSITA would understand the reference to polling “periodically” to imply “**predetermined polling frequencies**” because a frequency is inherent to a period.

Second, AAPA admits that polling at predetermined frequencies was well known in the prior art. AAPA admits the following:

- polling at predetermined frequency was known in the prior art including “commonly used” polling models. *Id.*, 3:11-14:

- “[t]he basic concept behind [prior art] polling models is that the client *periodically provides queries* to the host or server requesting updated messaging information.” *Id.*, 3:8-17.
- “*polling frequency*” can be “determined in existing polling based systems” and provides a specific polling frequency of “once per minute.” *Id.*, 4:20-23.
- in existing polling based systems, the “*polling frequency can be set* by knowing the number of expected clients and the available performance (potential requests/second) of the selected hardware.” *Id.*, 4:23-26; *see also id.*, 4:42-49 (describing that in the conventional polling system the GetUnreadAlerts is “the call initiated by the client to the server and is communicated to the server at the polling frequency”).

Therefore, *Dillon*-AAPA renders obvious that the client (*i.e.*, subscriber terminal) will contact the server (*i.e.*, e-mail server) at predetermined polling frequencies (*i.e.*, the reference to “periodically” in *Dillon* and the references to “polling frequency” that can be set in the AAPA).

6. Claim 3

[3.0] The method recited in claim 1 wherein said communications between said clients and said servers uses a SOAP protocol.

Dillon-AAPA renders obvious [3.0]. EX-1003, ¶¶102-104.

AAPA admits that the GetUnreadAlerts in the prior art Desktop Alert system used a SOAP call. EX-1001, 10:22-23 (“the client starts with the GetUnreadAlerts SOAP request as has been used in the prior art systems”). AAPA also admits that “SOAP” as recited in the claims was a “known computer communication protocol[].” EX-1022 at 15. Even if the SOAP protocol was not disclosed or suggested in *Dillon-AAPA*, using SOAP would have been an obvious matter of protocol choice. *Ex Parte Oliver*, No. 2009-005524, 2010 WL 3269925, at *7 (BPAI Aug. 12, 2010) (sustaining Examiner’s obviousness rejection rationale after acknowledging that prior art lacked the use of recited protocol but “which specific protocol to use would have been an obvious matter of protocol choice”).

Therefore, *Dillon-AAPA* renders obvious communications between clients (*i.e.*, subscriber terminals) and the server (*i.e.*, the e-mail server) using a SOAP protocol (which the AAPA admits was prior art and known).

7. Claims 4 and 5

[4.0] The method recited in claim 1 wherein said first communications between said clients and said servers uses a HTTP protocol and said second message from said client requesting unread alerts uses a SOAP protocol.

[5.0] The method recited in claim 4 wherein said first communications between said clients and said servers uses a HTTP GET protocol and said second message from said client requesting unread alerts uses a SOAP protocol.

Dillon-AAPA renders obvious [4.0] and [5.0]. EX-1003, ¶¶105-108.

First, as discussed with respect to [3.0], AAPA admits that SOAP was a “known computer communication protocol” in the prior art and an obvious matter of protocol choice. *See* Section X.A.6.

Second, with respect to claim 4, for the same reasons discussed in with respect to [1.4] the AAPA admits that HTTP was used in the prior art. *See* Section X.A.4.d. It would have been obvious to POSITA to use HTTP for the “**first communications between said clients and said servers.**” POSITA would have understood that it would have been an obvious matter of protocol choice to use HTTP for such communications. Furthermore, at the time a fine number of such protocols were known in the art, reducing the cost and risk of selection via experimentation.

Third, with respect to claim 5, the AAPA admits that HTTP GET was part of a “known computer communication protocol” in the prior art and an obvious matter of protocol choice. *See* Section X.A.2. Moreover, for the same reasons discussed with respect to [1.4], it would have been obvious to use HTTP GET to reduce the packet size of messages. *See* Section X.A.4.d.

8. Claim 6, 7, and 8

[6.0] The method recited in claim 1 wherein said communication proceeds according to an RPC protocol.

[7.0] The method recited in claim 6 further comprises a RPC/TCP protocol.

[8.0] The method recited in claim 6 further comprises a RPC/UDP protocol.

Dillon-AAPA renders obvious [6.0], [7.0], and [8.0]. EX-1003, ¶¶109-110.

AAPA admits that the RPC, RPC/UDP, and RPC/TCP protocols recited in the claims were “known computer communication protocols.” EX-1022 at 15. POSITA would have understood to use such known protocols for the communication to proceed. For instance, such known protocols make any resulting system compatible with existing systems, methods, and software. Using RPC (claim 6), RPC/TCP (claim 7), or RPC/UDP (claim 8) would have been an obvious matter of protocol choice. *See Oliver*, 2010 WL 3269925, at *7.

9. Claim 9

[9.0] The method of claim 1, wherein said alert message comprises: the date the message was received by said server, a time the message was received by said server, a date the message was sent to said client, a time the message was sent to said client, content of the message and combinations thereof.

Dillon-AAPA renders obvious [9.0]. EX-1003, ¶¶111-112.

Dillon describes the Date-Time Timestamps. EX-1005, 7:47-53 (“the time-stamps stored in the subscriber account memory 14 of the e-mail server 12 may be

numerical *time/date* values as noted above”). *Dillon* explains that the Date-Time Timestamp “indicat[es] a time when e-mail for the service subscriber was most recently *received* by the e-mail server.....[o]rdinarily, the e-mail notification that the front-end sends to a subscriber terminal for a particular e-mail service subscriber comprises a copy of the time-stamp for that e-mail service subscriber.” *Id.*, 3:47-55. *Dillon* includes claims that the “timestamp” is included in the “response sent to the client.” *Id.*, 17:43-44. POSITA would understand *Dillon*’s e-mails also include time/date that a “**message was sent to said client**” because e-mails include a “Received:” field for each computer in the e-mail’s path and the “Received:” data for the computer prior to . EX-1012, §3.6.7; *see also* EX-1023. To the extent *Dillon* does not disclose or suggest the “**sent**” information, POSITA would be motivated to modify *Dillon* to include this information in the “**alert message.**”

10. Claim 10

- a) [10.1] A system for providing alert messages to a plurality of clients using a polling-based network, said system comprising a plurality of clients,

Dillon-AAPA renders obvious [10.1]. EX-1003, ¶¶113-117.

First, for the reasons stated with respect to [1.1] and [1.2], *Dillon* discloses and/or renders obvious “**providing alert messages to a plurality of clients...said system comprising a plurality of clients.**” *See* Section X.A.4.a.-b.; EX-1005, FIG. 2, *infra*.

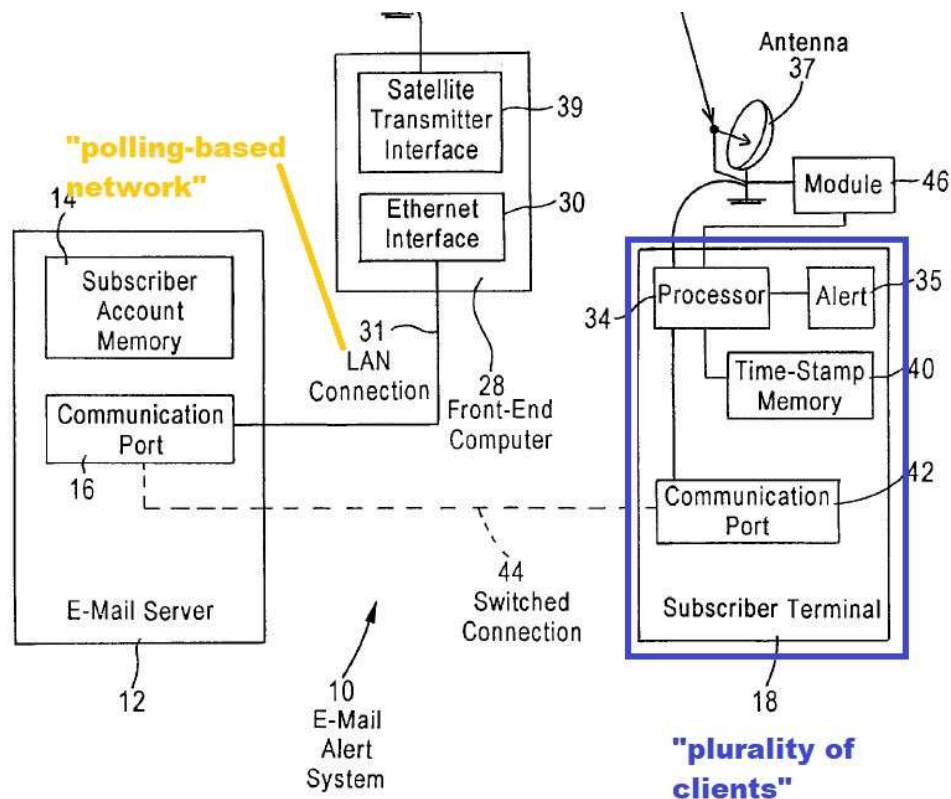


FIG. 2

EX-1005, FIG. 2

Second, Dillon discloses or suggests a “**polling-based network.**” *Dillon* discloses a “the front-end computer 28 includes an ethernet interface 30 providing a LAN connection 31 to the communication port 16 of the e-mail server 12, or any other suitable *polling means* for polling the e-mail server 12 to determine whether e-mail for an e-mail alert service subscriber has been received by the e-mail server 12.” *Id.*, 6:32-35. POSITA would understand the *Dillon*’s the ethernet interface 30, LAN connection 31, and communication port 16 to constitute a “**polling-based network.**”

Third, the AAPA discloses or suggests a “**polling-based network.**” AAPA admits that conventional polling models (the prior art Desktop Alert system being one example) were “widely adopted” in the prior art and particularly that polling models were “commonly used for checking email and other applications that require the timely updating of information such as stock prices and weather.” EX-1001, 3:14; 4:18-20.

Therefore, *Dillon*-AAPA renders obvious providing alert messages to a plurality of clients (*i.e.*, *Dillon*’s subscriber terminals) using a polling-based network (*i.e.*, the polling-based network of *Dillon* or AAPA), said system comprising a plurality of clients.

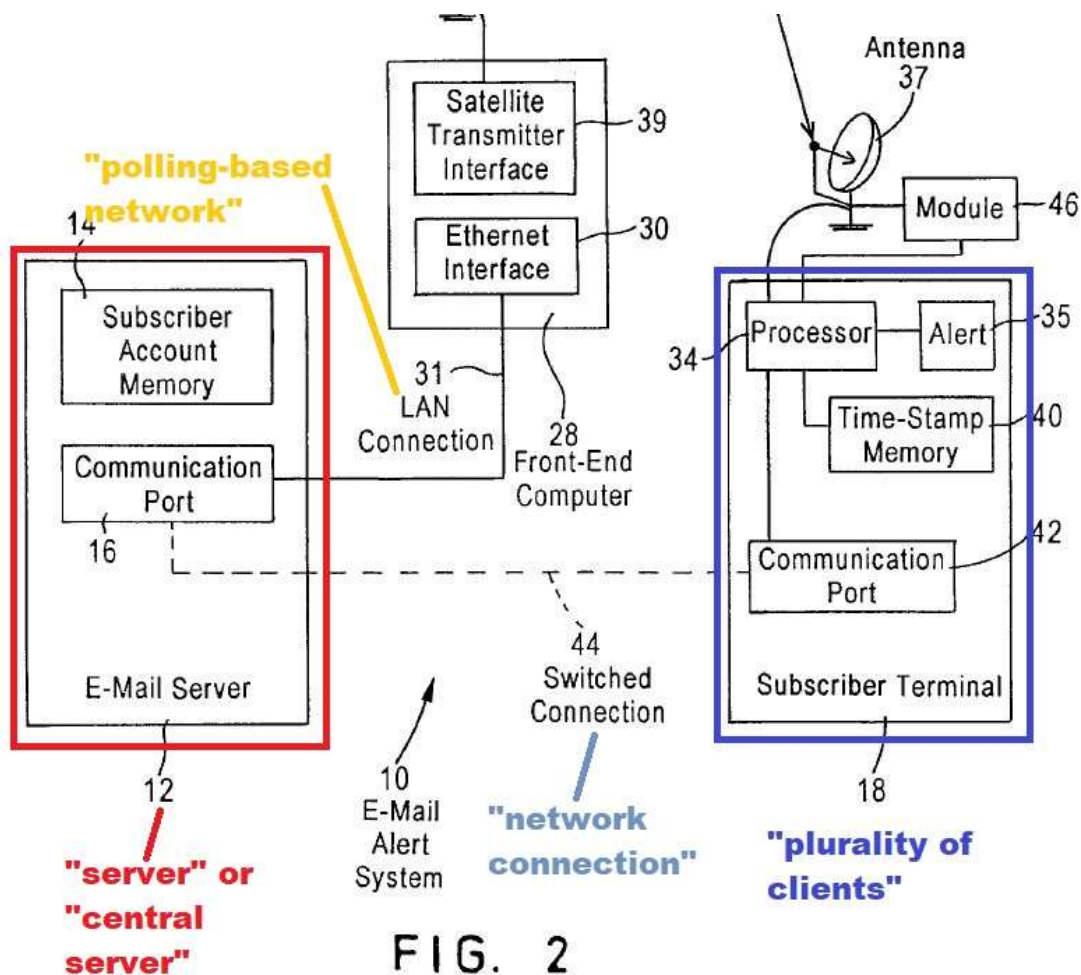
b) [10.2] at least one server in communication with said clients using a network connection,

Dillon-AAPA renders obvious [10.2]. EX-1003, ¶¶118-121.

First, for reasons similar to those stated with respect to [1.2], *Dillon* discloses or suggests “**at least one server in communication with said clients.**” In particular, *Dillon* describes an “e-mail server ... being accessible by the subscriber via subscriber terminal.” EX-1005, 5:33-34.

Second, *Dillon* discloses or suggests the communication between the server and clients “**using a network connection.**” *Dillon* describes that “e-mail may be downloaded via a link 44 (shown in phantom in FIG. 2) interconnecting the communication port 16 of the e-mail server 12 with the communication port 42 of

the subscriber terminal 18.” *Id.*, 9:41-45; *Id.*, FIG. 2 (labeling 44 as “Switched Connection”).



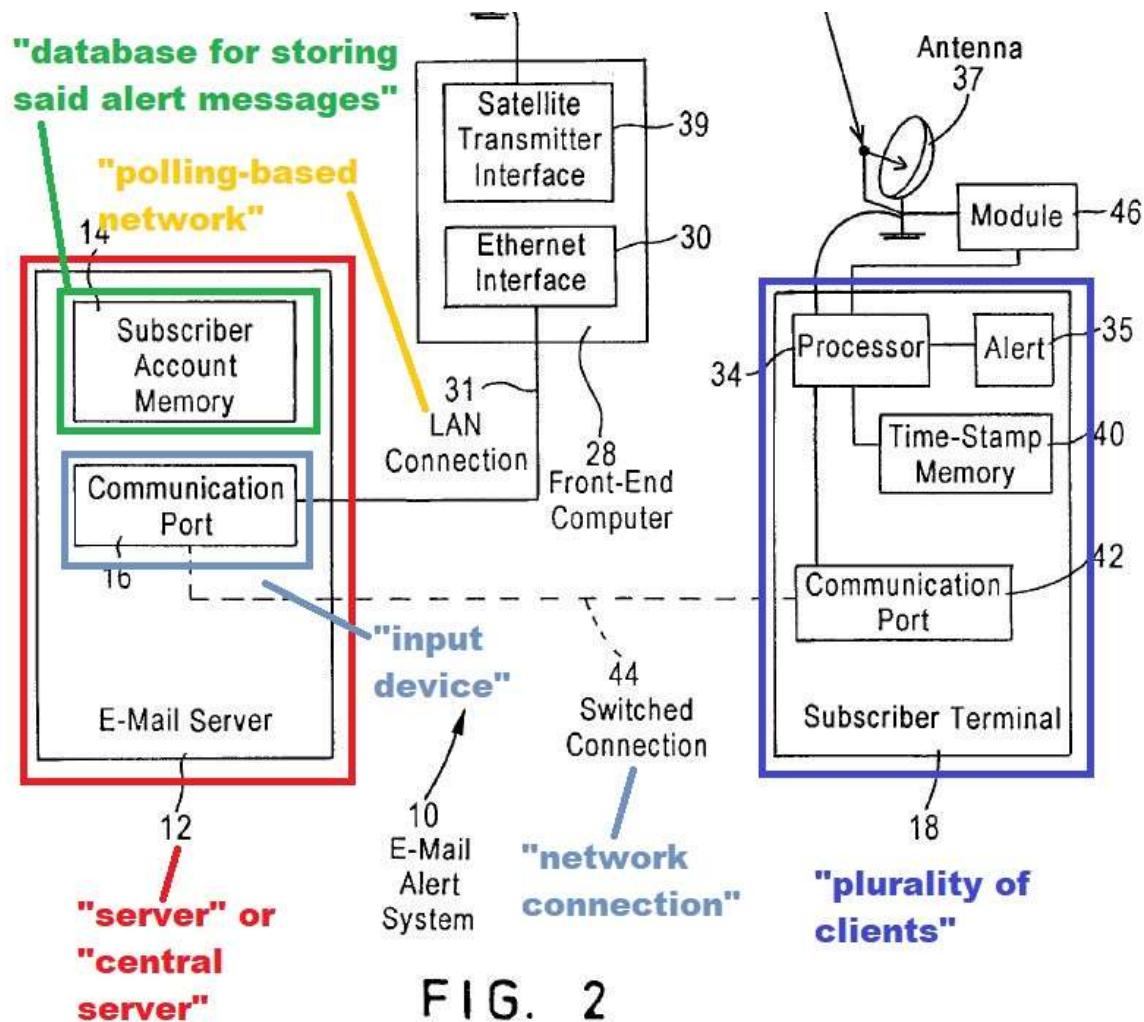
EX-1005, FIG. 2

Therefore, *Dillon-AAPA* renders obvious one server (*i.e.*, e-mail server) in communication with said clients (*i.e.*, subscriber terminal) using a network connection (*i.e.*, link and/or switched connection).

- c) **[10.3] said central server provided with input devices to receive alert messages, and a database, said database for storing said alert messages,**

Dillon-AAPA renders obvious [10.3]. EX-1003, ¶¶122-126.

First, Dillon discloses or suggests “**central server provided with input devices to receive alert messages.**” *Dillon* describes an “e-mail server 12 also includes a communication port 16 or other suitable communicating means for allowing the e-mail service subscriber to retrieve e-mail from the e-mail server 12 via a subscriber terminal 18 which generally is located remotely from the e-mail server 12.” EX-1005, 6:11-16.



EX-1005, FIG. 2

Second, Dillon discloses or suggests “a **database, said database for storing said alert messages.**” *Dillon* describes an “e-mail server includes a subscriber account memory....the subscriber account memory may also be used to store e-mail messages for e-mail service subscribers...” *Id.*, 3:26-35. *see also id.*, 5:66-6:9 (“As shown in FIG. 2, the server 12 includes a subscriber account memory 14 which is used to maintain information about e-mail accounts of the various subscribers to the

e-mail service. For example, when the e-mail server 12 receives e-mail for an e-mail service subscriber, the subscriber account memory 14 stores...the e-mail message itself.”). While *Dillon* does not use the word “database,” in the context of storing messages, POSITA would understand *Dillon* as disclosing or suggesting a **“database for storing said alert messages.”**

Third, the AAPA admits that the prior art Desktop Alert system made calls to a **“database for storing alert messages.”** *Id.*, 4:46-49 (referencing the “the resulting database calls or usage in the conventional polling system, such as the Desktop Alert system”). The AAPA further admits that, at the time of the invention, “industry standard database[s]” included SQL™, MySQL™, and Oracle™ databases”. *Id.*, 3:44-45; FIG. 12 (referencing “SQL Statement use, before”); 2:66-67 (describing FIG. 12 as “depicting SQL statement use, before the improvements”).

Therefore, *Dillon*-AAPA renders obvious said central server (*i.e.*, e-mail server) provided with input devices to receive alert messages (*i.e.*, communication port), and a database (*i.e.*, subscriber account memory), said database for storing said alert messages.

- d) **[10.4] and said processor having programming instructions to create and store an alert ID value[s] corresponding to alert messages stored in said database, and**

Dillon-AAPA renders obvious [10.4]. EX-1003, ¶¶127-130.

First, Dillon discloses or suggests a server that can “**create and store an alert ID value[s] corresponding to alert messages stored in said database.**”

Dillon describes a “time-stamp” that may “be assigned a value that uniquely identifies a particular e-mail message received by the e-mail server for an e-mail service subscriber.” EX-1005, 4:28-36. *Dillon* also describes that the “subscriber account memory 14 of the e-mail server 12 contains a time-stamp associated with each e-mail alert service subscriber...a time-stamp may be a number, such as a version number, that uniquely identifies a particular e-mail message received by the e-mail server 12 for the given e-mail alert service subscriber.” *Id.*, 7:15-29. *Dillon* describes that “the e-mail server 12 updates the time-stamp memory 14 for an e-mail service subscriber each time the e-mail server 12 receives e-mail for the e-mail service subscriber”). *Id.*, 8:34-39

Second, while *Dillon* does not use the phrases “**processor**” or “**programming instructions**,” POSITA would understand the e-mail server of *Dillon* to include both a processor and programming instructions as recited in the claim. For instance, *Dillon* describes a commercial embodiment of the e-mail server as a “a Sun Sparcstation running Software.com’s ‘Post Office’ e-mail server software.” A Sun SPARCstation was a computer workstation that included a “processor.” EX-1016 at 455. The reference to “e-mail server software” indicates that the e-mail server included “**programming instructions.**”

Therefore, *Dillon*-AAPA renders obvious renders obvious the processor (*i.e.*, a processor of the e-mail server, such as a processor of Sun SPARCstation) having programming instructions (*i.e.*, software) to create and store an alert ID value[s] (*i.e.*, Unique ID Timestamp) corresponding to alert messages (*i.e.*, e-mails) stored in said database (*i.e.*, subscriber account memory).

e) [10.5] said clients adapted to communicate with said server using an RPC communications protocol

Dillon-AAPA renders obvious [10.5]. EX-1003, ¶131-132.

In particular, the AAPA admits that the RPC protocol recited in the claims was a “known computer communication protocol[.]” EX-1022 at 15. POSITA would have understood to use such known protocols for the communication to proceed. For instance, such known protocols make any resulting system compatible with existing systems, methods, and software. Using RPC would have been an obvious matter of protocol choice. *See Oliver*, 2010 WL 3269925, at *7.

Therefore, *Dillon*-AAPA renders obvious clients (*i.e.*, subscriber terminals) adapted to communicate with the server (*i.e.*, the e-mail server) using an RPC communications protocol (which the AAPA admits is a known protocol).

- f) **[10.6] wherein said protocol further comprises a first triggering step comprises the use of a HTTP packet comprising a request for a Boolean value relating to whether there are new alert messages and said packet comprises approximately 20 bytes, and**

The claim language of [10.6] is substantively similar to [1.4]. For the same reasons discussed with respect to [1.4], *Dillon-AAPA* renders obvious [10.6]. See Section X.A.4.d.

- g) **[10.7] further communication from said client to said server requesting a Get Alert Messages using a SOAP communications protocol and**

Dillon-AAPA renders obvious [10.7]. EX-1003, ¶¶134-136.

AAPA admits that the GetUnreadAlerts call was used in the prior art Desktop Alert system and/or the “conventional polling system.” *Id.*, 4:39-49; *see also id.*, 4:41-42 (“the Desktop Alert web service GetUnreadAlerts call”); 4:46-48 (“critical GetUnreadAlerts call ... in the *conventional* polling system”); 4:50-51 (“[i]n the Desktop Alert system, each call to GetUnreadAlerts results in ...”); 5:59-61 (referencing the number of calls using the “*Existing* GetUnreadAlerts”); 6:7 (Table 1) (“Existing GetUnreadAlerts”); 6:11-24 (describing calls to GetUnreadAlerts in the “*Prior Art* System”). The specification also admits that the GetUnreadAlerts in the prior art Desktop Alert system used a SOAP call. *Id.*, 10:22-23 (“the client starts with the GetUnreadAlerts SOAP request as has been used in the prior art systems”).

Therefore, *Dillon*-AAPA renders obvious communication from the client (*i.e.*, the subscriber terminal) to said server (*i.e.*, the e-mail server) requesting a Get Alert Messages using a SOAP communications protocol (*i.e.*, the GetUnreadAlerts SOAP request in the AAPA).

h) [10.8] wherein said client is transmitted an Alert ID message value from said server to said client, and

Dillon-AAPA renders obvious [10.8]. EX-1003, ¶¶137-139.

Dillon discloses “the time-stamp stored in the subscriber account memory of the e-mail server is updated each time e-mail for the e-mail service subscriber is received by the e-mail server, and the polling means of the front-end polls the e-mail server periodically *to obtain the time-stamp for the e-mail service subscriber.*” EX-1005, 4:49-54. *see also id.*, 9:7-12 (“The comparison of the *time-stamp received from the e-mail server* 12 with the *most recently received* or ‘latest’ time-stamp stored in the time-stamp memory 40 *of the subscriber terminal* 18...”)

Therefore, *Dillon*-AAPA renders obvious a client (*i.e.*, subscriber terminal) is transmitted an Alert ID message value (*i.e.*, Unique ID Timestamp) from said server (*i.e.*, e-mail server) to said client (*i.e.*, subscriber terminal).

i) [10.9] if said Alert ID message value matches a stored message value, the client will further request from said server any alert messages to which it subscribes, and in response to said requests, said server transmits any unread alert messages and

Dillon-AAPA renders obvious [10.9]. EX-1003, ¶¶140-143.

Dillon discloses that “the subscriber terminal 18 determines, based on each e-mail notification it receives from the front-end computer 28, whether e-mail for a subscriber has been received by the e-mail server 12. To do so, the subscriber terminal 18 includes a time-stamp memory 40 for storing a ‘latest’ time-stamp which the subscriber terminal 18 most recently received in an e-mail notification from the front-end computer 28 for the e-mail alert service subscriber. The *comparison of the time-stamp received from the e-mail server 12 with the most recently received or ‘latest’ time-stamp stored in the time-stamp memory 40 of the subscriber terminal 18 is then made by the processor 34 or other suitable comparing means provided at the subscriber terminal 18.*” EX-1005, 8:66-9:12.

Dillon discloses “[i]f the received time-stamp differs from the latest time-stamp stored in the time-stamp memory 40, then ... the subscriber terminal 18 optionally can be suitably programmed to automatically dial in to the e-mail server 12 and retrieve e-mail messages for an e-mail alert service subscriber whenever the subscriber terminal 18 detects that the received time-stamp differs from the stored latest time10 stamp.” EX-1005, 9:13-27. While *Dillon* uses the word “differs” (and not “**matches**”), it would have been obvious to POSITA to modify *Dillon* to request an e-mail message when the Alert ID message value “**matches**” a stored message value.

Therefore, *Dillon-AAPA* renders obvious that if said Alert ID message value (*i.e.*, received time stamp) matches a stored message value (*i.e.*, stored latest time stamp), the client (*i.e.*, subscriber terminal) will further request from said server (*i.e.*, e-mail server) any alert messages (*i.e.*, e-mails) to which it subscribes, and in response to said requests, said server (*i.e.*, e-mail server) transmits any unread alert messages (*i.e.*, unread e-mails).

- j) **[10.10] said unread alert message are in instant message format and upon the client's reception of said unread alert messages said client opens a window on a display panel at said client, and**

The claim language of [10.10] is similar to [1.11]. For the same reasons discussed with respect to [1.11], *Dillon-AAPA* renders obvious [10.10]. *See* Section X.A.4.k.

- k) **[10.11] wherein said communications protocol comprises protocols for the transmission of messages over the internet.**

The claim language of [10.11] is similar to [1.12]. For the same reasons discussed with respect to [1.12], *Dillon-AAPA* renders obvious [10.11]. *See* Section X.A.4.l.

11. Claim 11

[11.0] The system of claim 10, wherein said server further comprising an application programming interface controls the administrator interface and provides at least one instruction to the administrator interface for creating said alert messages and transmitting said alert messages to the user contact devices.

Dillon-AAPA renders obvious [11.0]. EX-1003, ¶¶146-148.

The AAPA discloses or suggests an “**instruction....for creating said alert messages and transmitting said alert messages to the user contact devices.**”

AAPA admits that the prior art Desktop Alert system included an “[e]mbedded web-based html editor for ... alert creation using HTML and Custom scripts.” EX-1001, 1:42-43. The specification further admits that the Desktop Alert system included a feature where “[a]lerts may be created by the user from pre-defined and customized alert templates for rapid alert publication.” *Id.*, 2:7-8. POSITA would understand AAPA to include such features using “**an application programming interface [that] controls the administrator interface.**”

Therefore, *Dillon*-AAPA renders obvious server (*i.e.*, e-mail server) that comprises an application programming interface controls the administrator interface (*i.e.*, embedded web-based editor) and provides at least one instruction to the administrator interface for creating said alert messages (*i.e.*, the instructions for creating an alert, including the pre-defined/customized alert templates) and

transmitting said alert messages to the user contact devices (*i.e.*, which is indicated by the reference to rapid alert publication).

12. Claim 12

[12.0] The system of claim 10, further comprising display devices at said clients and at said server wherein said client displays, display [sic: displays] said alter [sic: alert] messages.

The claim language of [12.0] is similar to [1.11]. EX-1003, ¶¶149-150. For the same reasons discussed with respect to [1.11], *Dillon*-AAPA renders obvious **“display devices at said clients...wherein client displays...said [alert] messages.”** See Section X.A.4.k. *Dillon* also depicts clients (*i.e.*, subscriber terminals) having displays. EX-1005, FIG. 1. *Dillon* also describes a commercial embodiment as having a “Sun Sparcstation” as an e-mail server. *Id.*, 5:21-23. POSITA would have understood a Sun Sparcstation as disclosing or suggesting **“display devise[] at...said server.”**

Therefore, *Dillon*-AAPA renders obvious the system comprising display devices at said clients (*i.e.*, depicted in FIG. 1) and at said server (*i.e.*, the display of the Sun SPARCstation) wherein said client displays the alert messages (*i.e.*, e-mails).

13. Claim 13

[13.0] The system of claim 10, wherein said database is an SQL™ database, MySQL™ database, an Oracle™ database, another industry standard database, or combinations thereof.

Dillon-AAPA renders obvious [13.0]. EX-1003, ¶¶151-152.

AAPA admits that, at the time of the invention, “*industry standard* database[s]” existed including “SQL™, MySQL™, and Oracle™” databases. EX-1001, 3:44-45; FIG. 12 (referencing “SQL Statement use, *before*”); 2:66-67 (describing FIG. 12 as “depicting SQL statement use, *before* the improvements”). Moreover, claim 13 itself concedes that these databases are all “**industry standard.**” POSITA would have been motivated to use an “industry standard database” including “SQL™, MySQL™, and Oracle™.” There is nothing arguably inventive to select such conventional, industry-standard choices for the claimed “**database.**”

B. Ground 2: *Dillon-Brady*-AAPA Render Obvious Claims 1-13

1. Overview of *Brady*

Brady is directed to “automatically polling for data changes in an on-demand database service environment.” EX-1008, ¶[0009]. *Brady* describes an “HTTP client...for sending and receiving HTTP messages to and from an HTTP server” and specifically discloses “incoming HTTP requests.” *Id.*, ¶[0028]. Such “requests” are sent by a “polling client ... that ... calls or queries the polling server.” *Id.*, ¶¶[0053]-[0057].

2. Reasons to Combine *Dillon*-AAPA with *Brady*

As previously discussed, a POSITA would be motivated to combine *Dillon* and AAPA. See Section X.A.3. A POSITA would be further motivated to combine *Dillon*-AAPA with *Brady*. EX-1003, ¶156. *Brady* is directed to “automatically *polling* for data changes in an on-demand database service environment.” EX-1008, ¶[0009]. Both *Dillon* and the AAPA are directed to polling systems. See Section X.A.3. Thus, *Dillon*, *Brady*, and AAPA are analogous prior art to the '765 Patent and are in the same field of *polling* systems. A POSITA would have been motivated to utilize conventional aspects (such as use of HTTP packets comprising a request taught in *Brady*) in any implementations of network details left unspecified by *Dillon*-AAPA. These implementation details and/or modifications would have involved combining prior art elements (*i.e.*, *Dillon*-AAPA polling system) according to known methods (*i.e.*, the use of HTTP packets comprising a request as taught by *Brady*) to yield the predictable result (*i.e.*, polling for and retrieval of retrieving unread e-mails with reduced database usage and increased polling frequency). See *KSR*, 550 U.S. at 416. A POSITA would have had reasonable expectations of success given that *Dillon*, AAPA, and *Brady* all build upon conventional and standardized networking technologies and protocols employed for their intended purposes. For instance, a POSITA would have little doubt that using an HTTP

packet comprising a request would be a reliable and efficient form given the widely-used and pervasive nature of the HTTP standards. A POSITA would thus have had a reasonable expectation of success in combining *Dillon*-AAPA with *Brady*. *Regents*, 903 F.3d at 1291.

3. Claims 1-13

As discussed above, *Dillon*-AAPA renders claims 1-13 obvious. *See* Section X.A. In particular, as discussed above in with respect to [1.4] and [10.6], *Dillon*-AAPA discloses or suggests that a “**triggering step comprises the use of a HTTP packet comprising a request.**” *See* Sections X.A.4.d. and X.A.10.f. But to the extent Patent Owner argues or the Board finds that *Dillon*-AAPA does not disclose or suggest an HTTP packet comprising a request, *Brady* describes an “HTTP client...for sending and receiving HTTP messages to and from an HTTP server” and specifically discloses “incoming HTTP requests.” EX-1008, ¶[0028]. Such “requests” are sent by a “polling client ... that ... calls or queries the polling server.” *Id.*, ¶¶[0053]-[0057]. *Brady* also discloses a “**triggering**” step because the use of the HTTP packet the step results in a function or process being initiated. *Id.*, ¶[0055] (“If the id is new, the polling client should cause the page to be refreshed.”).

Therefore, *Dillon-Brady* -AAPA renders obvious a triggering step comprises the use of a HTTP packet comprising a request. The above modification of *Dillon*-AAPA does not affect the remainder of the analysis in Section X.A (Ground 1),

which applies equally to this Ground. Thus, *Dillon- Brady*-AAPA renders obvious claims 1-13 for the reasons discussed in Section X.A (Ground 1) and those discussed above. EX-1003, ¶¶157-158.

C. Ground 3: *Dillon-Douglas*-AAPA Render Obvious Claims 1-13

1. Overview of *Douglas*

Douglas discloses method and system for “managing the reception of messages” including “one or more client nodes and a server node, which stores messages associated with the one or more client nodes. EX-1006, ¶¶[0014]. “The messages can be electronic mail messages (emails) or Instant Messages (IMs).” *Id.* A message account (associate with a client node) polls the server node to check for the arrival of new messages (such as Instant Messages) associated with the message account at the server node. *Id.*, ¶[0015]. When there are new messages at the server node, the polling results show the arrival of new messages, and the first client node can retrieve and read the new messages. *Id.*

2. Reasons to Combine *Dillon*-AAPA with *Douglas*

As previously discussed, a POSITA would be motivated to combine *Dillon* and AAPA. *See* Section X.A.3. A POSITA would be further motivated to combine *Dillon*-AAPA with *Douglas*. EX-1003, ¶160. *Douglas* is directed to “first client node....polling...a server node....to check for the arrival of a first message.” automatically *polling* for data changes in an on-demand database

service environment.” EX-1006, Abstract. Both *Dillon* and the AAPA are directed to polling systems. *See* Section X.A.3. Thus, *Dillon*, *Douglas*, and AAPA are analogous prior art to the ’765 Patent and are in the same field of *polling* systems. Both *Dillon* and the AAPA are silent on the format of messages. But *Douglas* discloses “Instant Messages (IMs)” as an alternative message format to “electronic mail messages (emails).” EX-1006, ¶[0014]. POSITA would have been motivated to utilize conventional aspects (such as the Instant Message format taught by *Douglas*) in any implementations of network details left unspecified by *Dillon*-AAPA. These implementation details and/or modifications would have involved combining prior art elements (*i.e.*, *Dillon*-AAPA polling system) according to known methods (*i.e.*, Instant Messaging taught by *Douglas*) to yield the predictable result (*i.e.*, transmitting and displaying messages). *See KSR*, 550 U.S. at 416. A POSITA would have had reasonable expectations of success given that *Dillon*, AAPA, and *Douglas* all build upon conventional and standardized networking technologies and protocols employed for their intended purposes. For instance, a POSITA would have little doubt a message in Instant Message format would be a reliable and efficient form given the widely-used and pervasive nature of Instant Messages. A POSITA would thus have had a reasonable expectation of success in combining *Dillon*-AAPA with *Douglas*. *Regents*, 903 F.3d at 1291.

3. Claims 1-13

As discussed above, *Dillon*-AAPA renders claims 1-13 obvious. *See* Section X.A. In particular, as discussed above in with respect to [1.11] and [10.10], *Dillon*-AAPA discloses or suggests that a “**said alert messages are in instant message format.**” *See* Sections X.A.4.k. and X.A.10.j. But to the extent Patent Owner argues or the Board finds that *Dillon*-AAPA does not disclose or suggest an alert messages in instant message format, *Douglas* describes that “messages can be electronic mail messages (emails) or Instant Messages (IMs).” EX-1006, ¶¶[0014].

Therefore, *Dillon-Douglas*-AAPA renders obvious alert messages being in instant message format. The above modification of *Dillon*-AAPA does not affect the remainder of the analysis in Section X.A. (Ground 1), which applies equally to this Ground. Thus, *Dillon-Douglas*-AAPA renders obvious claims 1-13 for the reasons discussed in Section X.A. (Ground 1) and those discussed above. EX-1003, ¶¶161-162.

D. Ground 4: *Dillon-Brady-Douglas*-AAPA Render Obvious Claims 1-13

1. Reasons to Combine *Dillon-Brady*-AAPA with *Douglas*

As previously discussed, a POSITA would be motivated to combine (i) *Dillon* and AAPA; (ii) *Dillon*, *Brady*, and AAPA; and (iii) *Dillon*, *Douglas*, and AAPA. *See* Sections X.A.3., X.B.2., X.C.2; EX-1003, ¶163. For those same reasons, a POSITA would be motivated to combine *Dillon-Brady*-AAPA with *Douglas*. For

instance, *Dillon*, *Brady*, *Douglas*, and AAPA are all directed to polling systems. *See* Sections X.A.3., X.B.2., X.C.2. For the same reasons as discussed, a POSITA would have a reasonable expectation of success given to combine *Dillon-Brady-AAPA* with *Douglas*. *See id.* For instance, a POSITA would have little doubt a message in Instant Message format of *Douglas* would be a reliable and efficient form given the widely-used and pervasive nature of Instant Messages. A POSITA would thus have had a reasonable expectation of success in combining *Dillon-Brady-AAPA* with *Douglas*. *Regents*, 903 F.3d at 1291.

2. Claims 1-13

As referenced above, *Dillon-Brady-AAPA* renders claims 1-13 obvious. *See* Section X.C.3. As discussed above in with respect to [1.11] and [10.10] (which are referenced in the *Dillon-Brady-AAPA* analysis), *Dillon-Brady-AAPA* discloses or suggests that a “**said alert messages are in instant message format.**” *See* Sections X.A.5.k. and X.A.11.j. But to the extent Patent Owner argues or the Board finds that *Dillon-Brady-AAPA* does not disclose or suggest an alert messages in instant message format, *Douglas* describes that “messages can be electronic mail messages (emails) or Instant Messages (IMs).” EX-1006, ¶¶[0014].

Therefore, *Dillon-Brady-Douglas-AAPA* renders obvious alert messages being in instant message format. The above modification of *Dillon-Brady-AAPA* does not affect the remainder of the analysis in Section X.C. (Ground 3), which

applies equally to this Ground. Thus, *Dillon-Brady-Douglas-AAPA* renders obvious claims 1-13 for the reasons discussed in Section X.C. (Ground 3) and those discussed above. EX-1003, ¶¶164-165.

* * *

XI. CONCLUSION

Petitioner requests institution of an IPR and cancelation of the Challenged Claims.

Dated: January 8, 2024

Respectfully submitted,
By: /s/ Brian M. Koide
Brian M. Koide (Reg. No. 41,123)
Lead Counsel

XII. CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing argument section of this Petition for *Inter Partes* Review of U.S. Patent No. 9,172,765 (Sections I and III-XI) contains, as measured by the word-processing system used to prepare this paper, 13,601 words. This word count does not include the items excluded by 37 C.F.R. § 42.24 as not counting towards the word limit.

XIII. CLAIMS APPENDIX (U.S. PAT. NO. 9,172,765)⁹

[1.1] An improved method for the dissemination of an alert message to a plurality of personal computers using a network communications protocol,

[1.2] said computers comprising clients, each said clients subscribing to a central server, said server comprising an alert message cache, said server adapted to receive and store alert messages, and

[1.3] said server adapted to generate and store alert ID numbers, said method comprising,

[1.4] a first triggering step comprises the use of a HTTP packet comprising a request for a Boolean value relating to whether there are new alert messages and said packet comprises approximately 20 bytes,

[1.5] a second step wherein said client contacts the central server and requests the value of said alert ID number in said memory, said value reflecting the number of alert messages received by said central server,

[1.6] a third step wherein said server transmits said value to said client,

[1.7] a fourth step wherein said client compares a stored value at the client with the value of the transmitted alert ID number, and

⁹ Numbering in brackets added. A copy of the Claims Appendix is also submitted as EX-1025.

[1.8] if said transmitted value exceeds said stored value, a fifth step wherein said client transmits a second message to said server and requests for the server to transmit all unread alert messages in a GetUnreadAlerts request from said server that are specific to said client, and

[1.9] if there are unread alert messages specific to said client in said server, a sixth step wherein said server sends to client all unread alert messages specific to the client, and,

[1.10] a further step wherein said client updates the stored value in its memory to reflect the number of additional unread alert messages transmitted from said server to said client in response to said second message, and

[1.11] wherein, upon the reception of said unread alert messages in said sixth step, said alert messages are in instant message format and said client opens a window on a display panel to display said alert messages,

[1.12] wherein said communications protocol comprises protocols for the transmission of messages over the internet.

[2.0] The method recited in claim 1 wherein said client will contact said server at predetermined polling frequencies.

[3.0] The method recited in claim 1 wherein said communications between said clients and said servers uses a SOAP protocol.

[4.0] The method recited in claim 1 wherein said first communications between said clients and said servers uses a HTTP protocol and said second message from said client requesting unread alerts uses a SOAP protocol.

[5.0] The method recited in claim 4 wherein said first communications between said clients and said servers uses a HTTP GET protocol and said second message from said client requesting unread alerts uses a SOAP protocol.

[6.0] The method recited in claim 1 wherein said communication proceeds according to an RPC protocol.

[7.0] The method recited in claim 6 further comprises a RPC/UDP protocol.

[8.0] The method recited in claim 6 further comprises a RPC/TCP protocol.

[9.0] The method of claim 1, wherein said alert message comprises: the date the message was received by said server, a time the message was received by said server, a date the message was sent to said client, a time the message was sent to said client, content of the message and combinations thereof.

[10.1] A system for providing alert messages to a plurality of clients using a polling-based network, said system comprising a plurality of clients,

[10.2] at least one server in communication with said clients using a network connection,

[10.3] said central server provided with input devices to receive alert messages, and a database, said database for storing said alert messages,

[10.4] and said processor having programming instructions to create and store an alert ID value corresponding to alert messages stored in said database, and

[10.5] said clients adapted to communicate with said server using an RPC communications protocol

[10.6] wherein said protocol further comprises a first triggering step comprises the use of a HTTP packet comprising a request for a Boolean value relating to whether there are new alert messages and said packet comprises approximately 20 bytes, and

[10.7] further communication from said client to said server requesting a Get Alert Messages using a SOAP communications protocol and

[10.8] wherein said client is transmitted an Alert ID message value from said server to said client, and

[10.9] if said Alert ID message value matches a stored message value, the client will further request from said server any alert messages to which it subscribes, and in response to said requests, said server transmits any unread alert messages and

[10.10] said unread alert message are in instant message format and upon the client's reception of said unread alert messages said client opens a window on a display panel at said client, and

[10.11] wherein said communications protocol comprises protocols for the transmission of messages over the internet.

[11.0] The system of claim 10, wherein said server further comprising an application programming interface controls the administrator interface and provides at least one instruction to the administrator interface for creating said alert messages and transmitting said alert messages to the user contact devices.

[12.0] The system of claim 10, further comprising display devices at said clients and at said server wherein said client displays, display said alter messages.

[13.0] The system of claim 10, wherein said database is an SQL™ database, MySQL™ database, an Oracle™ database, another industry standard database, or combinations thereof.

XIV. CERTIFICATE OF SERVICE

I hereby certify that on January 8, 2024, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 9,172,765 and supporting exhibits to be served via express mail on Desktop Alert Inc. at the following correspondence address of record as listed on Patent Center:

Andrew C. Aitken
P.O. Box 1810
Wheaton, MD 20915

Further, I caused a true and correct courtesy copy of the foregoing Petition and supporting exhibits to be served via e-mail on Patent Owner's counsel of record in the Litigation:

Joseph Farco
Norris McLaughlin, P.A.
7 Times Square, 21st Floor
New York, NY 10036
jfarco@norris-law.com

Andrew C. Aitken
IP Law Leaders PLLC
6701 Democracy Blvd., Suite 555
Bethesda, MD 20817
acaitken@aitkenlawoffices.com

By: /s/ Brian M. Koide /
Brian M. Koide (Reg. No. 41,123)