Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 92045

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

Cedar Point Partners, LLC

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature	/Robert Paul Morris/	Date (Optional)	
Name	Robert Paul Morris		
Title	Manager - Cedar Point Partners, LLC		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☐ Total of forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Robert	Paul	Morris	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	Raleigh	State/Province	NC	Country of Residence	US

Mailing Address of Inventor:

Address 1	712 Latta St		
Address 2			
City	Raleigh	State/Province	NC
Postal Code	27607	Country	US

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	92045		
Email Address	pcaldwell@thecaldwellfirm.com	Add Email	Remove Email

Application Information:

Title of the Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION		
Attorney Docket Number	DR0149-CIP	Small Entity Status Claimed	☒
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)		Suggested Figure for Publication (if any)	

Filing By Reference :

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☐ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	◉ Customer Number	○ US Patent Practitioner	○ Limited Recognition (37 CFR 11.9)
Customer Number	92045		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.
When referring to the current application, please leave the application number blank.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation in part of	13624906	2012-09-22
Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation in part of	13647144	2012-10-08
Prior Application Status	Pending		Remove

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation in part of	13626635	2012-09-25
Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation in part of	13624940	2012-09-23
Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation in part of	13654647	2012-10-18
Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation in part of	13716156	2012-12-16
Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation in part of	13716160	2012-12-16
Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation in part of	13716159	2012-12-16
Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation in part of	13716158	2012-12-16

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button.

Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(d). When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)[i] the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(h)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Remove

Application Number	Country[i]	Filing Date (YYYY-MM-DD)	Access Code[i] (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button.

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

Authorization to Permit Access:

☐ Authorization to Permit Access to the Instant Application by the Participating Offices

If checked, the undersigned hereby grants the USPTO authority to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the World Intellectual Property Office (WIPO), and any other intellectual property offices in which a foreign application claiming priority to the instant patent application is filed access to the instant patent application. See 37 CFR 1.14(c) and (h). This box should not be checked if the applicant does not wish the EPO, JPO, KIPO, WIPO, or other intellectual property office in which a foreign application claiming priority to the instant patent application is filed to have access to the instant patent application.

In accordance with 37 CFR 1.14(h)(3), access will be provided to a copy of the instant patent application with respect to: 1) the instant patent application-as-filed; 2) any foreign application to which the instant patent application claims priority under 35 U.S.C. 119(a)-(d) if a copy of the foreign application that satisfies the certified copy requirement of 37 CFR 1.55 has been filed in the instant patent application; and 3) any U.S. application-as-filed from which benefit is sought in the instant patent application.

In accordance with 37 CFR 1.14(c), access may be provided to information concerning the date o f filing this Authorization.

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Applicant 1

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Clear

◉ Assignee	○ Legal Representative under 35 U.S.C. 117	○ Joint Inventor
○ Person to whom the inventor is obligated to assign.	○ Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor :

If the Applicant is an Organization check here. ☒

Organization Name	Cedar Point Partners, LLC

Mailing Address Information For Applicant:

Address 1	712 Latta St		
Address 2			
City	Raleigh	**State/Province**	NC
Country	US	Postal Code	27607
Phone Number		Fax Number	
Email Address			

Additional Applicant Data may be generated within this form by selecting the Add button.

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not subsitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication . An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1			
Address 2			
City		**State/Province**	
Country		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button.

Signature:

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Signature	/Patrick E. Caldwell/			Date (YYYY-MM-DD)	2014-05-09
First Name	Patrick	Last Name	Caldwell	Registration Number	44580

Additional Signature may be generated within this form by selecting the Add button.

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

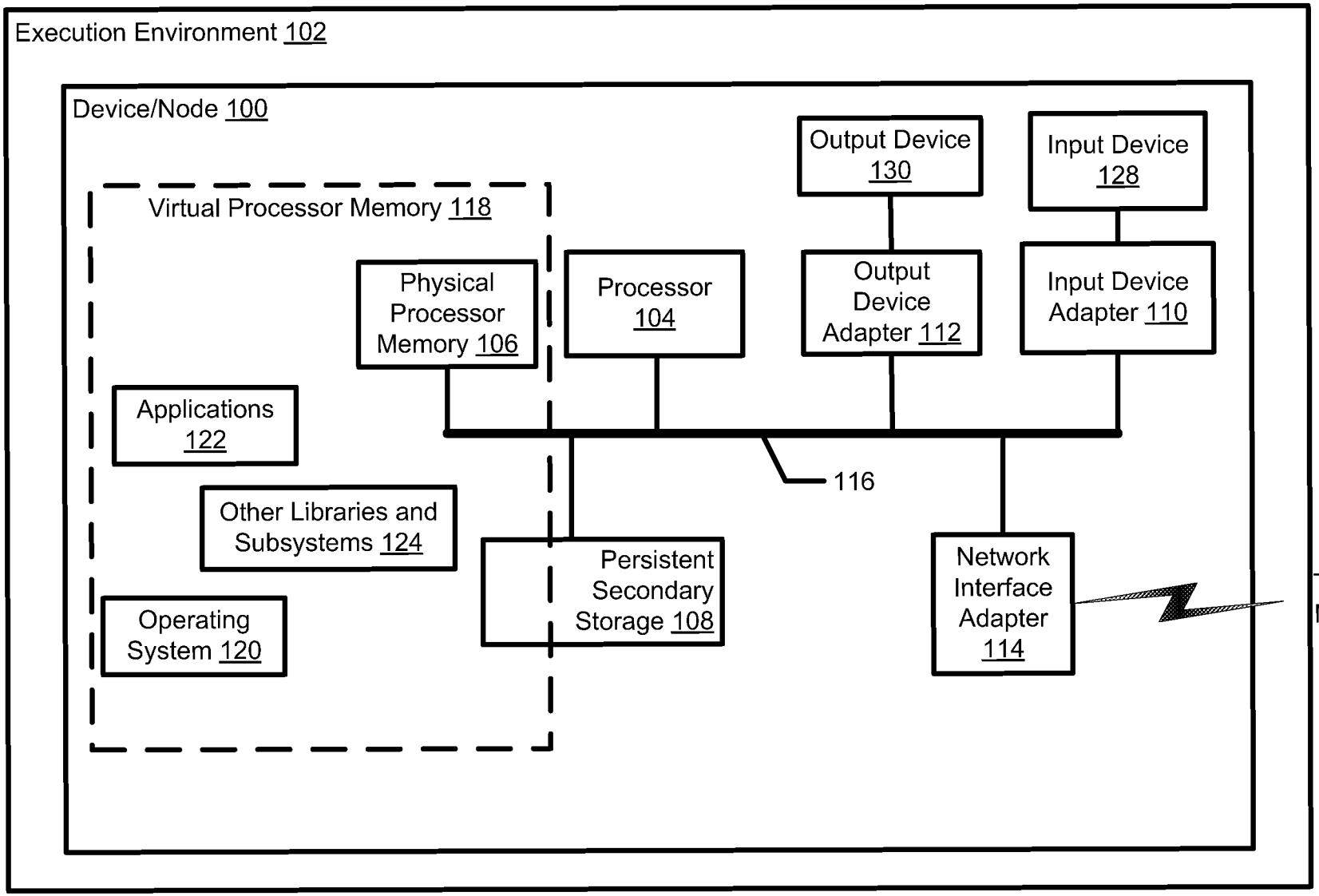

Execution Environment 102
Device/Node 100
Virtual Processor Memory 118
Physical Processor Memory 106
Processor 104
Output Device 130
Input Device 128
Output Device Adapter 112
Input Device Adapter 110
Applications 122
Other Libraries and Subsystems 124
Operating System 120
Persistent Secondary Storage 108
116
Network Interface Adapter 114
To/From Network

FIG. 1

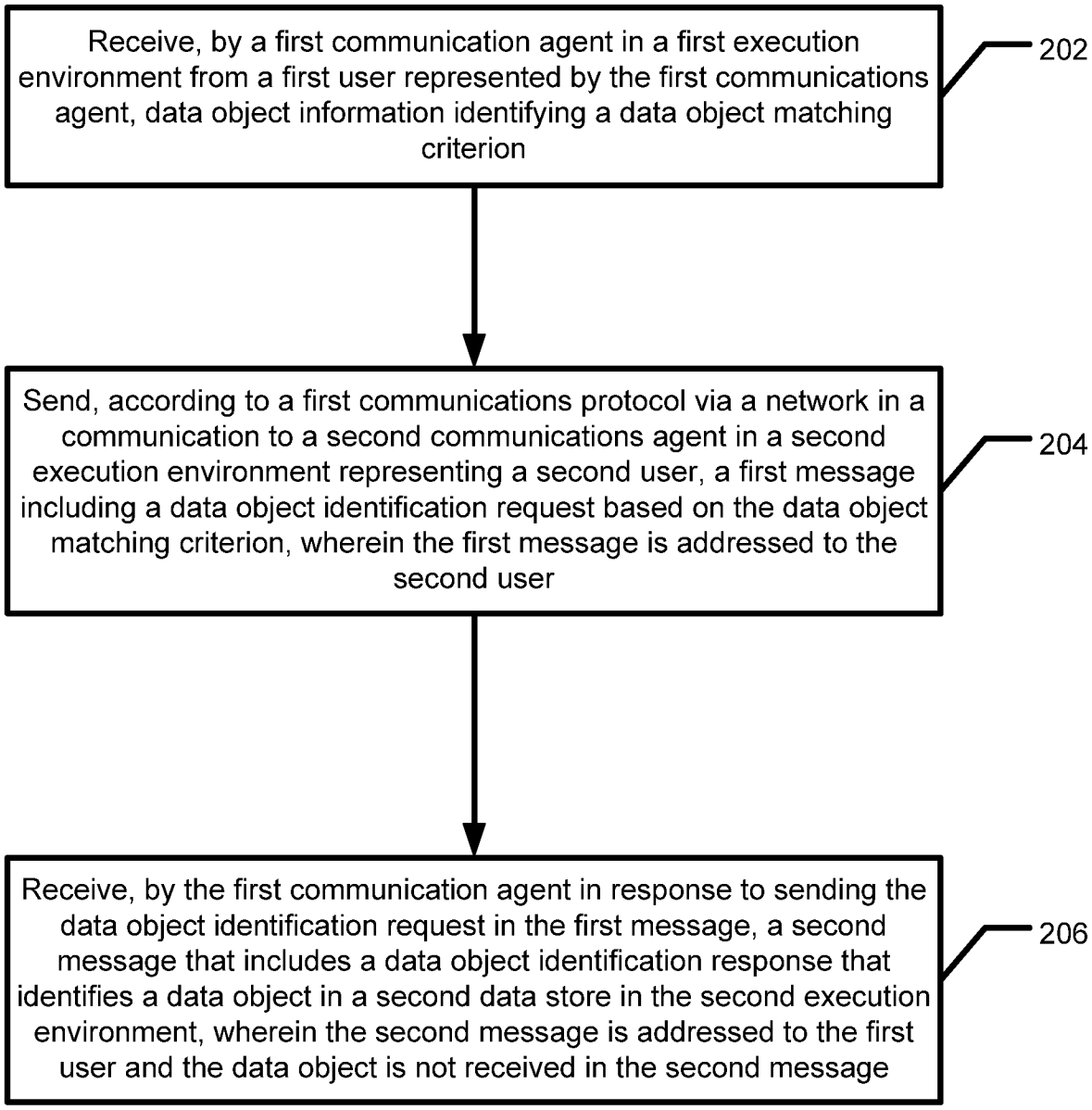

Receive, by a first communication agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion
202
Send, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user
204
Receive, by the first communication agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message
206

FIG. 2A

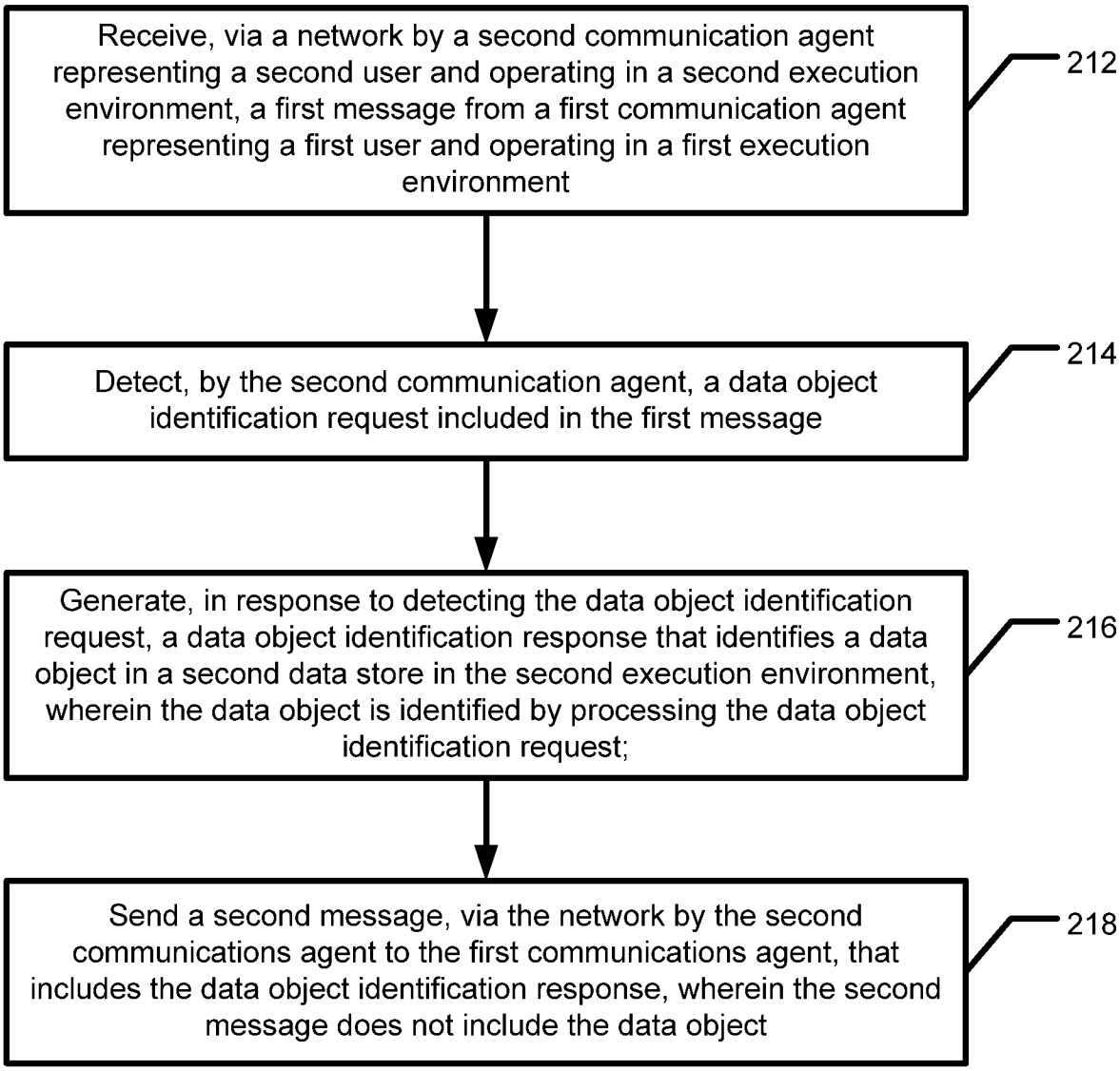

Receive, via a network by a second communication agent representing a second user and operating in a second execution environment, a first message from a first communication agent representing a first user and operating in a first execution environment
212
Detect, by the second communication agent, a data object identification request included in the first message
214
Generate, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request;
216
Send a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object
218

FIG. 2B

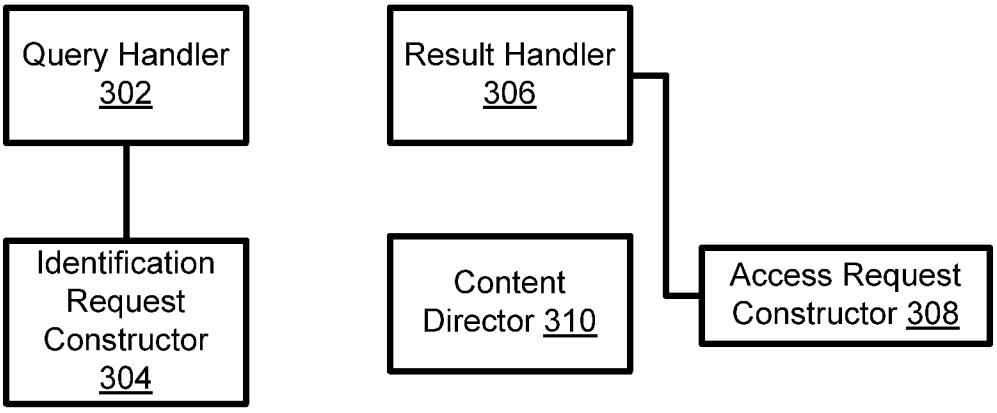

Query Handler 302
Result Handler 306
Identification Request Constructor 304
Content Director 310
Access Request Constructor 308

FIG. 3A

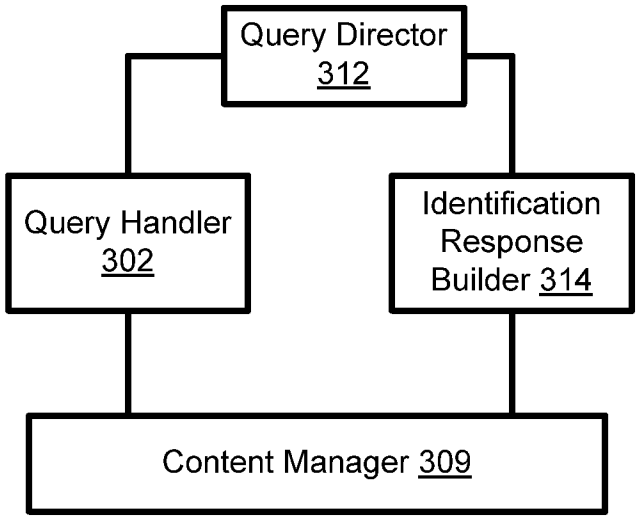

Query Director 312
Query Handler 302
Identification Response Builder 314
Content Manager 309

FIG. 3B

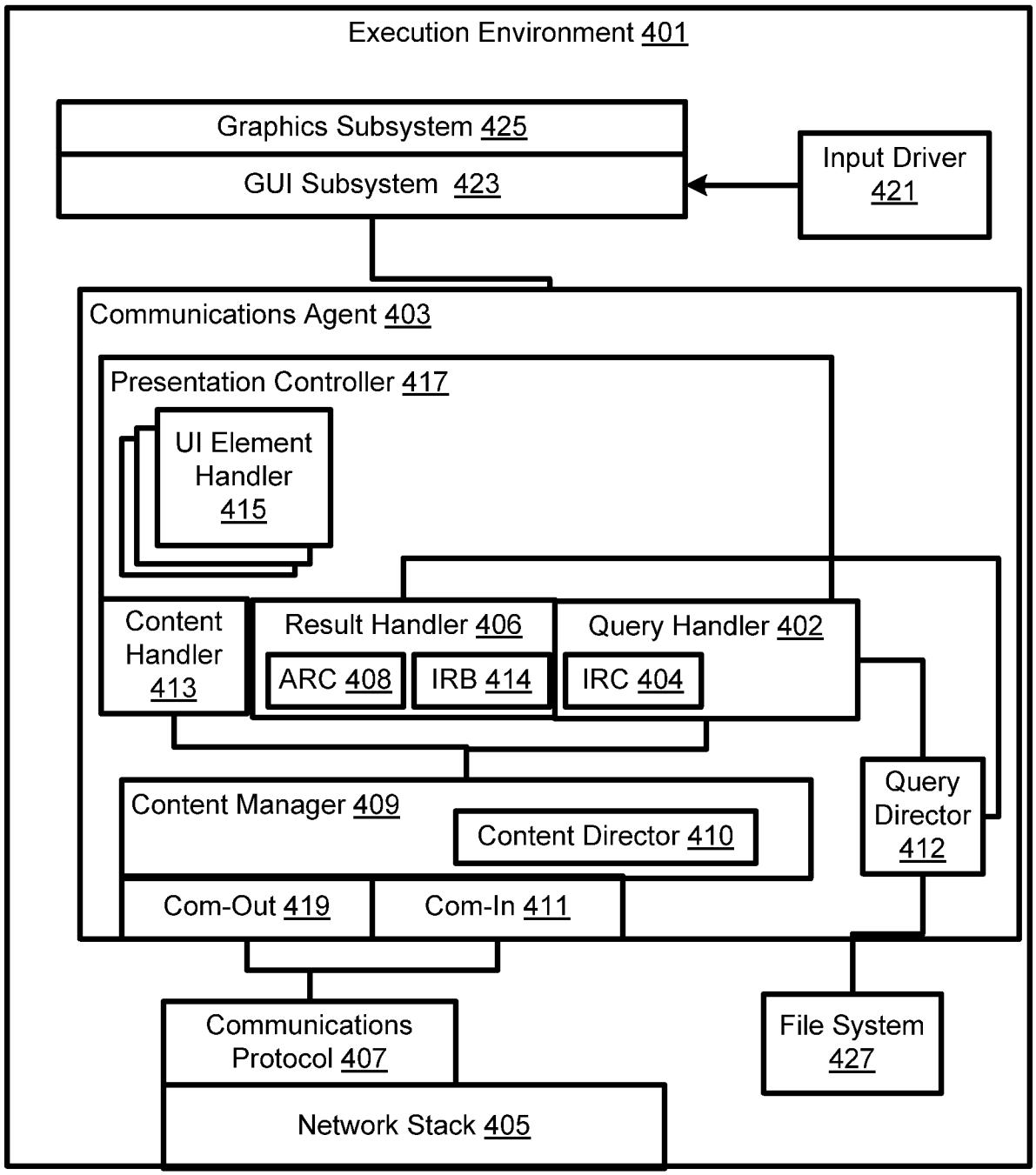

Execution Environment 401
Graphics Subsystem 425
GUI Subsystem 423
Input Driver 421
Communications Agent 403
Presentation Controller 417
UI Element Handler 415
Content Handler 413
Result Handler 406
ARC 408
IRB 414
Query Handler 402
IRC 404
Content Manager 409
Content Director 410
Query Director 412
Com-Out 419
Com-In 411
Communications Protocol 407
Network Stack 405
File System 427

FIG. 4

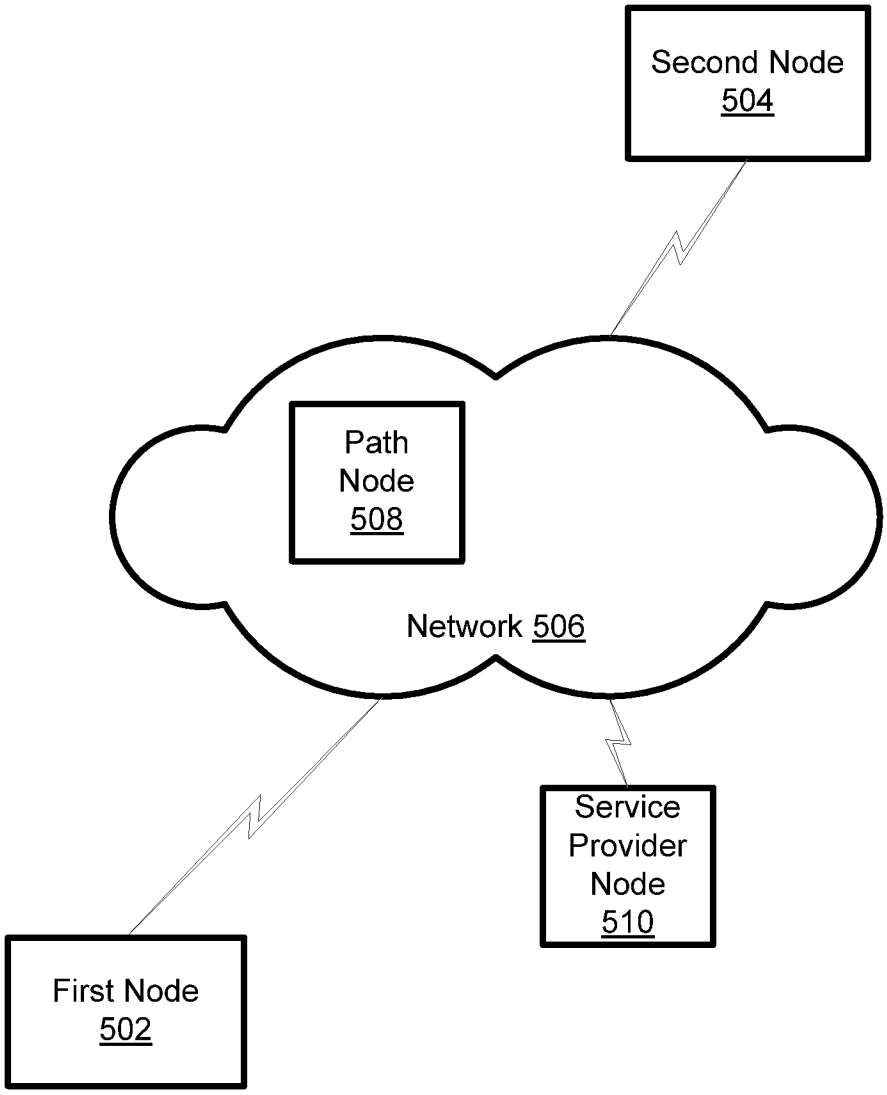

Second Node
504
Path
Node
508
Network 506
Service
Provider
Node
510
First Node
502

FIG. 5

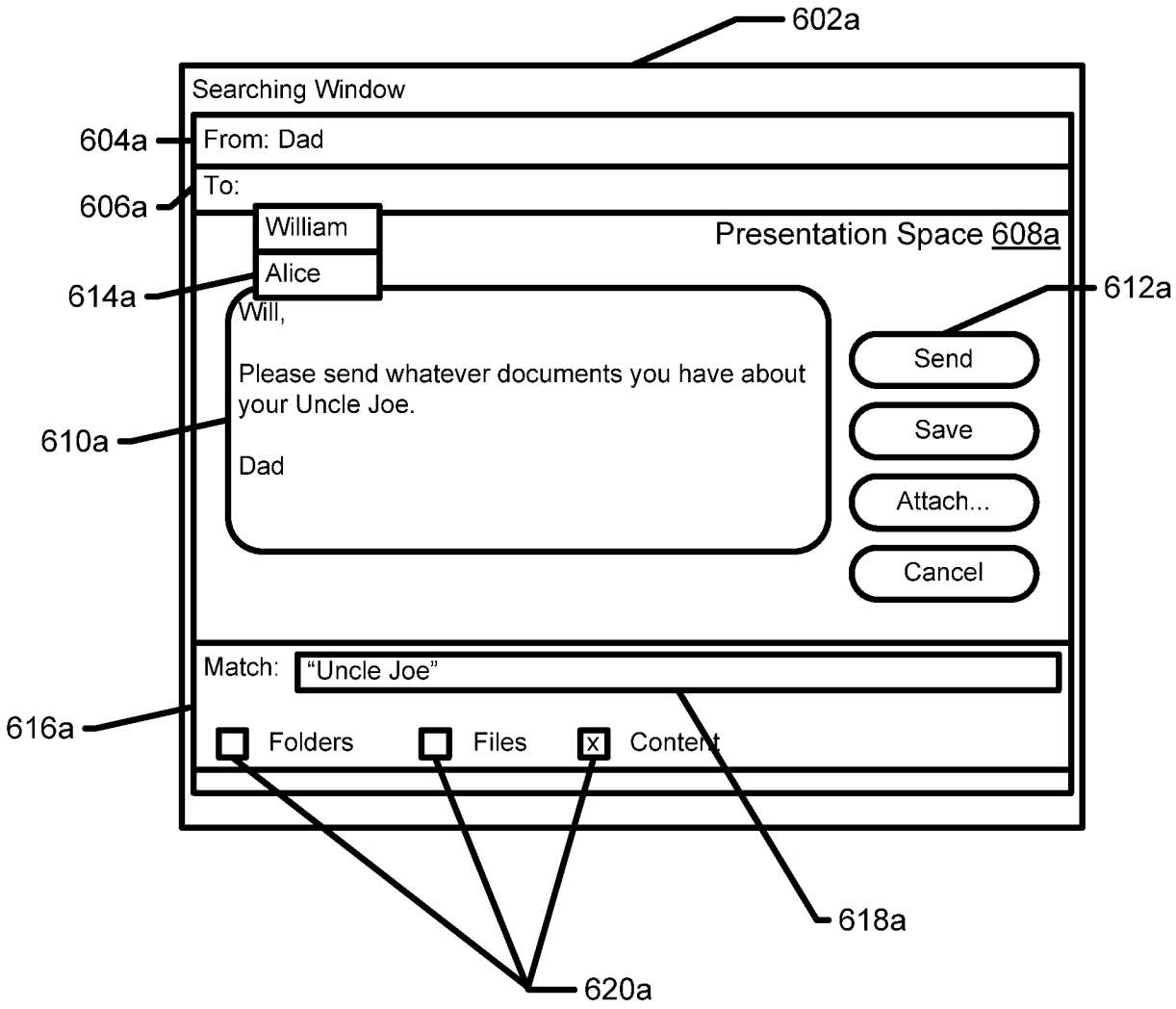

602a
Searching Window
604a
From: Dad
606a
To:
William
Alice
Presentation Space 608a
614a
612a
Will,
Please send whatever documents you have about your Uncle Joe.
610a
Dad
Send
Save
Attach...
Cancel
Match:
"Uncle Joe"
616a
Folders
Files
Content
618a
620a

FIG. 6A

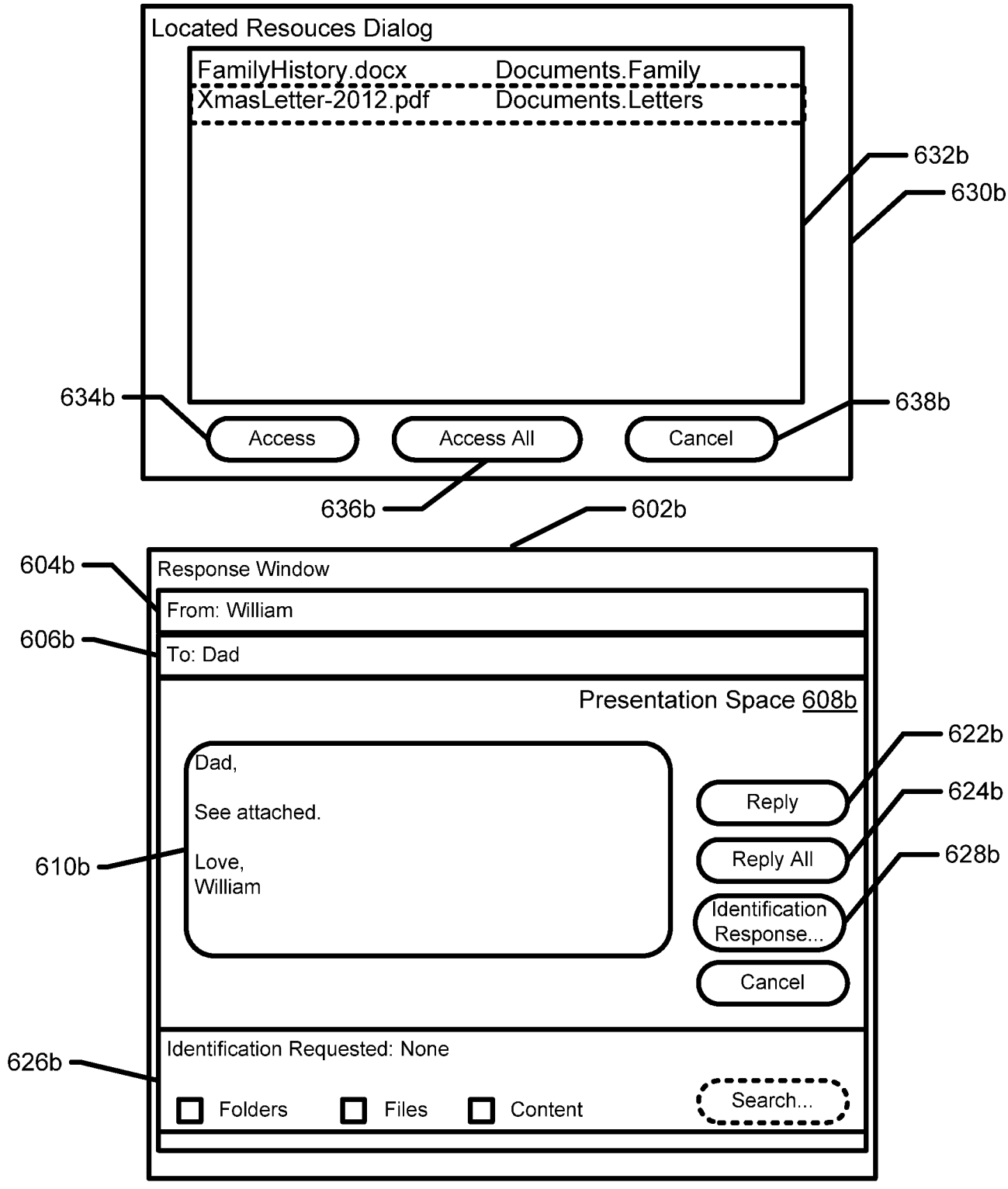

Located Resouces Dialog
FamilyHistory.docx
Documents.Family
XmasLetter-2012.pdf
Documents.Letters
632b
630b
634b
638b
Access
Access All
Cancel
636b
602b
604b
Response Window
From: William
606b
To: Dad
Presentation Space 608b
622b
624b
628b
610b
Dad,
See attached.
Love,
William
Reply
Reply All
Identification Response...
Cancel
626b
Identification Requested: None
Folders
Files
Content
Search...

FIG. 6B

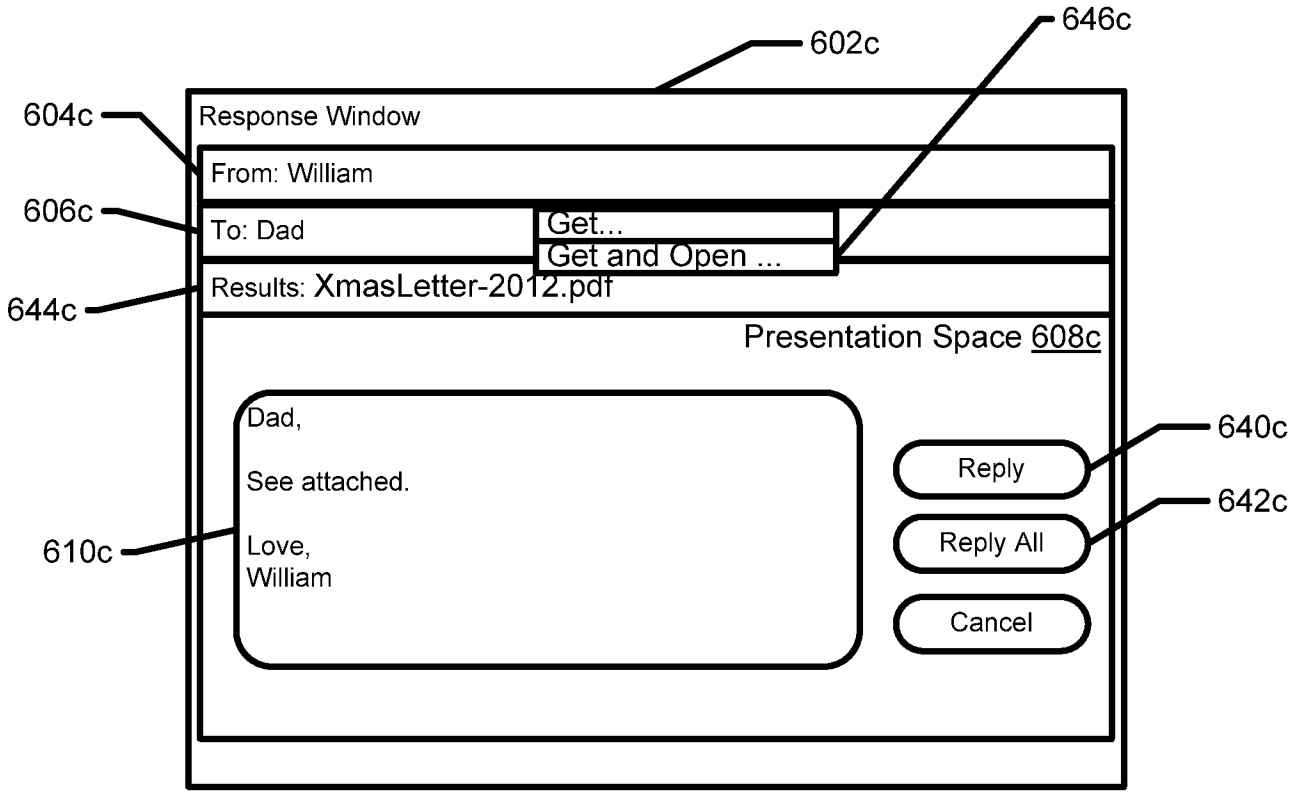

602c
646c
604c
Response Window
From: William
606c
To: Dad
Get...
Get and Open ...
644c
Results: XmasLetter-2012.pdf
Presentation Space 608c
Dad,
See attached.
Love,
William
610c
Reply
640c
Reply All
642c
Cancel

FIG. 6C

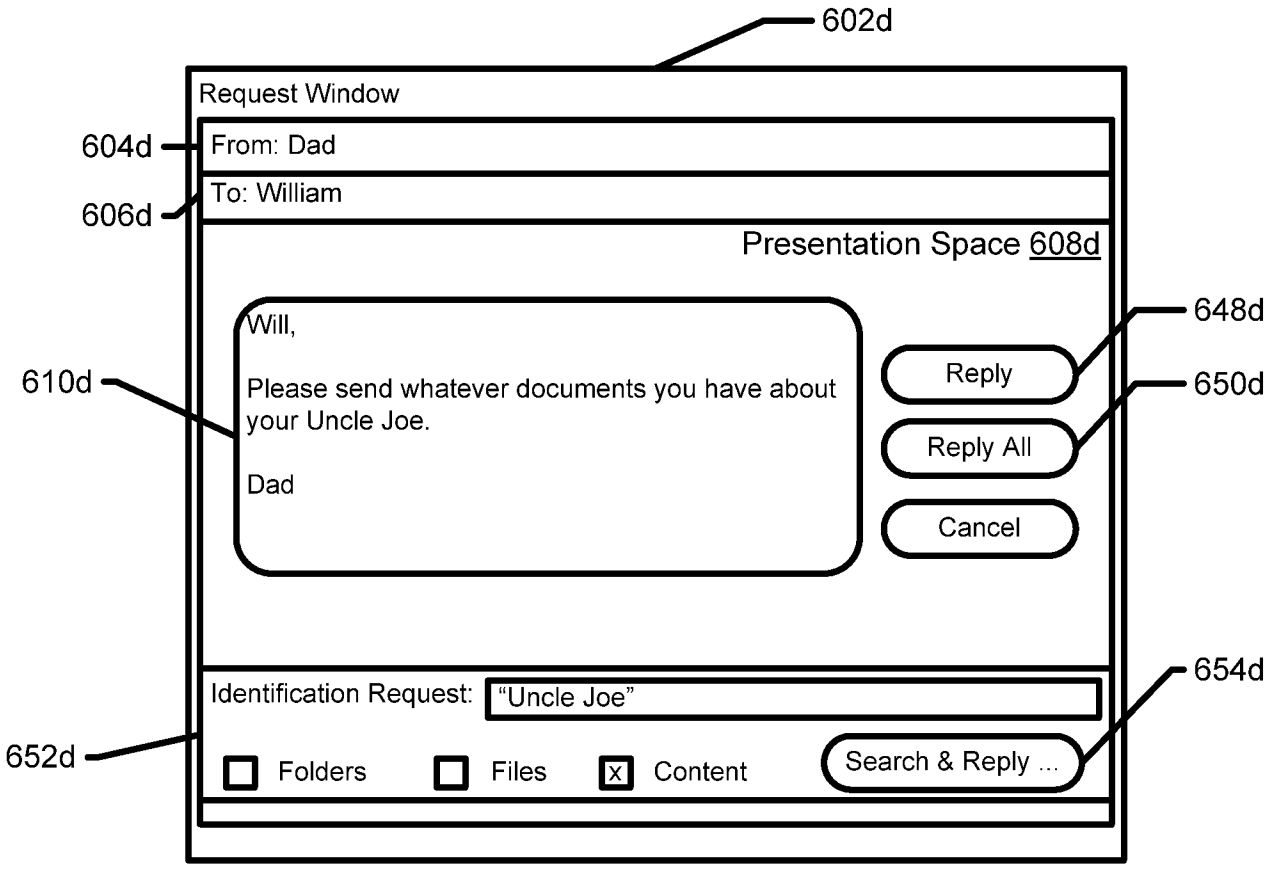

602d
Request Window
604d
From: Dad
606d
To: William
Presentation Space 608d
610d
Will,
Please send whatever documents you have about your Uncle Joe.
Dad
Reply
Reply All
Cancel
648d
650d
654d
Identification Request:
"Uncle Joe"
652d
Folders
Files
Content
Search & Reply ...

FIG. 6D

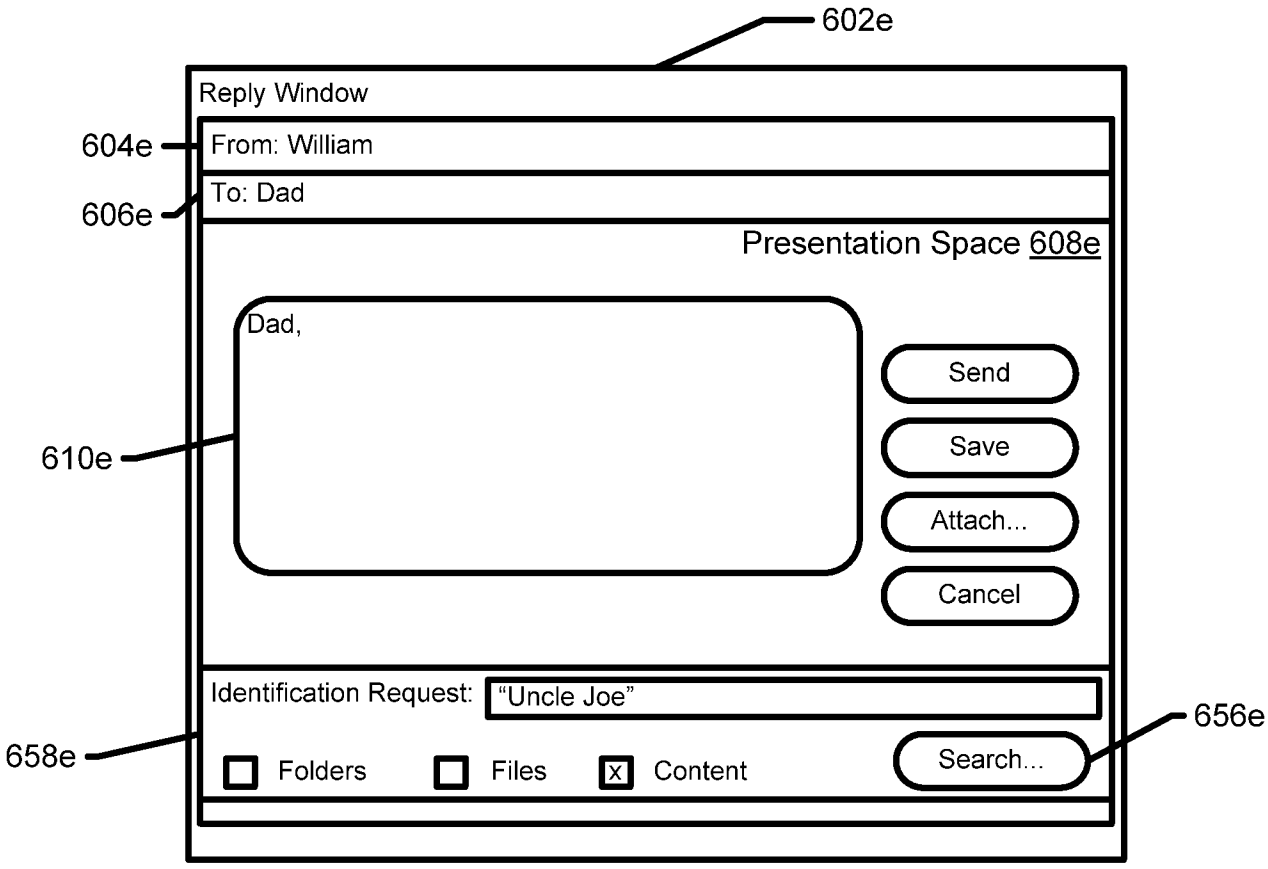

602e
Reply Window
604e
From: William
606e
To: Dad
Presentation Space 608e
Dad,
610e
Send
Save
Attach...
Cancel
Identification Request:
"Uncle Joe"
658e
656e
Folders
Files
X
Content
Search...

FIG. 6E

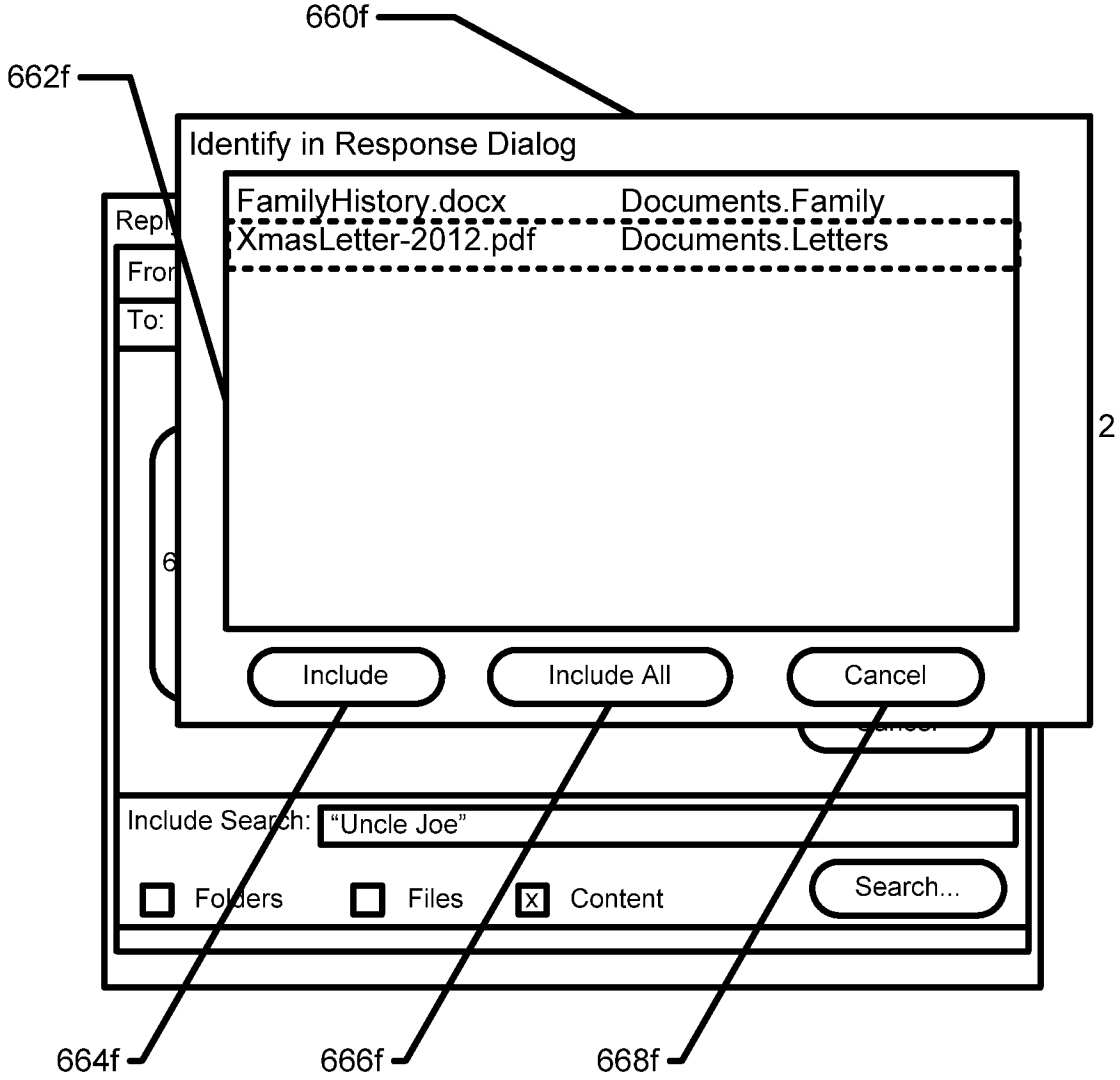

660f
662f
Identify in Response Dialog
FamilyHistory.docx Documents.Family
XmasLetter-2012.pdf Documents.Letters
Include
Include All
Cancel
Include Search: "Uncle Joe"
Folders
Files
Content
Search...
664f
666f
668f

FIG.6F

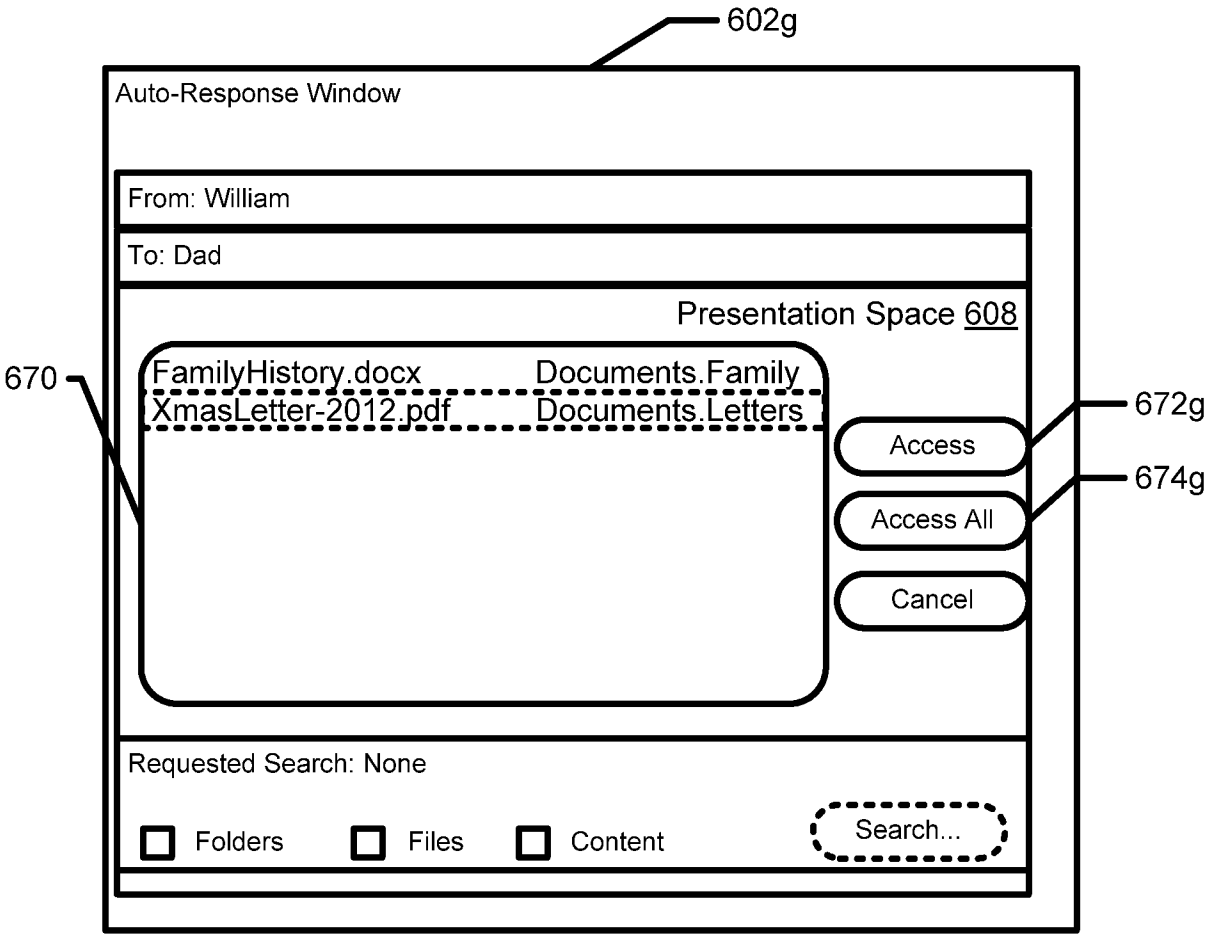

602g
Auto-Response Window
From: William
To: Dad
Presentation Space 608
670
FamilyHistory.docx Documents.Family
XmasLetter-2012.pdf Documents.Letters
Access
Access All
Cancel
672g
674g
Requested Search: None
Folders
Files
Content
Search...

FIG. 6G

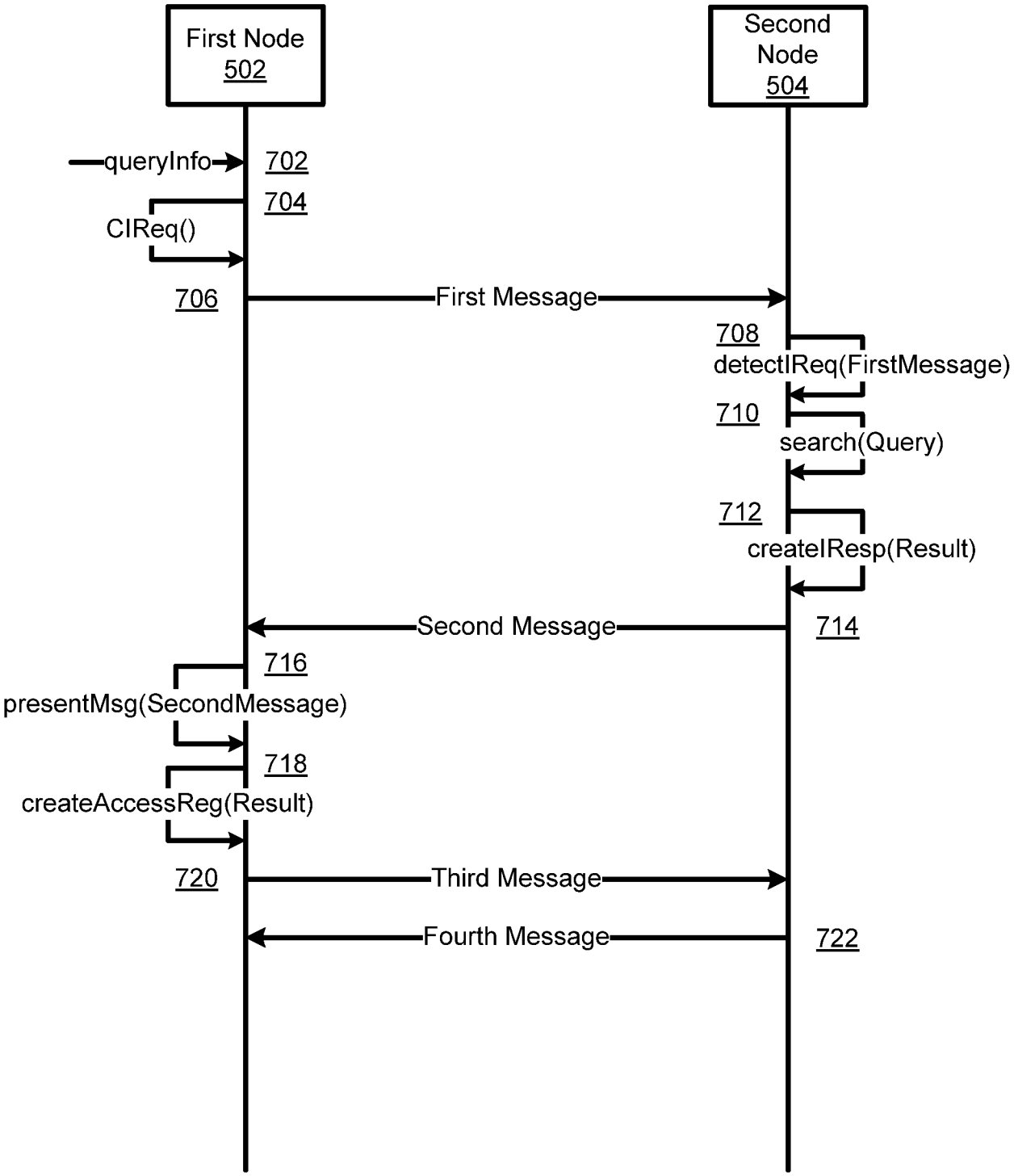

First Node
502
Second
Node
504
queryInfo
702
704
CIReq()
706
First Message
708
detectIReq(FirstMessage)
710
search(Query)
712
createIResp(Result)
Second Message
714
716
presentMsg(SecondMessage)
718
createAccessReg(Result)
720
Third Message
Fourth Message
722

FIG. 7

800a

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="--OoOT----toOo--------"

--OoOT----toOo--------
Content-Type: text/plain
Hi, this is the first message
--OoOT----toOo--------
Content-Type: application/id-request
Accept:  image/*
<id-request>
  <and>
    <criterion attribute="query" type="keyword" match-expression="'Uncle Joe'" />
    <criterion attribute="scope" resource-type="content" />
  </and>
</id-request>
--OoOT----toOo--------
```

814a, 812a, 804a, 806a, 810a, 808a, 802a

Fig. 8A

800b

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="--toOo----OoOt--------"

--toOo----OoOt---------
Content-Type: text/plain
Dad,

See attached.

Love,
William
--toOo----OoOt---------
Content-Type: application/query-respponse

FILE://Documents/Family/FamilyHistory.docx
FILE://Documents/Letters/XmasLetter-2012.pdf
--toOo----OoOt---------
```

802b, 806b, 804b

FIG. 8B

METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION

RELATED APPLICATIONS

[0001] The present application claims priority to U.S. Application No. 13/624,906 filed 09/22/2012 and entitled " METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION," U.S. Application No. 13/647,144 filed 10/08/2012 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR EXCHANGING PRESENTATION DATA BASED ON A COMMUNICATION," U.S. Application No. 13/626,635 filed 09/25/2012 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR SHARING A DATA OBJECT IN A DATA STORE VIA A COMMUNICATION," U.S. Application No. 13/624,940 filed 09/23/2012 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A REFERENCE IN A COMMUNICATION TO A REMOTE DATA OBJECT," U.S. Application No. 13/654,647 filed 10/18/2012 and entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR CONSTRAINING A DATA EXCHANGE REQUESTED IN A COMMUNICATION," U.S. Application No. 13/716,156 filed 12/16/2012 and entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A SEARCH QUERY EXCHANGED VIA A COMMUNICATIONS PROTOCOL," U.S. Application No. 13/716,160 filed 12/16/2012 and entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR ACCESSING A SERVICE VIA A PROXY COMMUNICATIONS AGENT," U.S. Application No. 13/716,159 filed 12/16/2012 and entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR BROWSING VIA A COMMUNICATIONS AGENT," and U.S. Application No. 13/716,158 filed 12/16/2012 and entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A REQUEST VIA A COMMUNICATIONS AGENT," the entire contents of each are herein incorporated by reference.

[0002] This application is related to the following commonly owned U.S. Patent Applications: Application No. 12/833,016 (Docket No 0158) filed on 2010/07/09, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR REFERENCING AN ATTACHMENT IN A COMMUNICATION"; and

[0003] Application No. 12/833,014 (Docket No 0155) filed on 2010/07/09, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A REQUEST FOR A RESOURCE IN A COMMUNICATION".

BACKGROUND

[0004] While receiving attachments in a network communication, such as email, is common, most attachments received are unsolicited or unrequested. To request an attachment, a user typically sends the request as voice and/or text data in a voice and/or text message heard and/or read by another user. The request may be as vague or as specific as the language used by the requesting user. The other user must interpret the request and find a file or other data object that seems to match the request. The other user in many cases locates a "matching" resource using a program or application other than the communications agent that received the request. For example, a search program may be used to search a hard-drive by the other user. The other user must enter the search criteria.

[0005] Accordingly, there exists a need for methods, systems, and computer program products for processing a data object identification request in a communication.

SUMMARY

[0006] The following presents a simplified summary of the disclosure in order to provide a basic understanding to the reader. This summary is not an extensive overview of the disclosure and it does not identify key/critical elements of the invention or delineate the scope of the invention. Its sole purpose is to present some concepts disclosed herein in a simplified form as a prelude to the more detailed description that is presented later.

[0007] Methods and systems are described for processing a data object identification request in a communication. In one aspect, the method includes receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. The method further includes sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. The method still further includes receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message.

[0008] Also, a system for processing a data object identification request in a communication is described that includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion; sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object

matching criterion, wherein the first message is addressed to the second user; and receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message.

[0009] Further, a system for processing a data object identification request in a communication is described. The system includes a query handler component for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. The system further includes an identification request constructor component for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. The system still further includes a result handler component for receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. The system also includes a processor, wherein at least one of the query handler component, the identification request constructor component, and the result handler component includes an instruction that is executed by the processor during operation of the system.

[0010] Methods and systems are described for processing a data object identification request in a communication. In one aspect, the method includes receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. The method further includes detecting, by the second communications agent, a data object

identification request included in the first message. The method still further includes generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. The method additionally includes sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. Performing at least one the preceding actions comprising the method includes execution of an instruction by a processor.

[0011] Also, a system for processing a data object identification request in a communication is described that includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment; detecting, by the second communications agent, a data object identification request included in the first message; generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request; and sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object.

[0012] Further, a system for processing a data object identification request in a communication is described. The system includes a content manager component for receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. The system further includes a query handler component for detecting, by the second communications agent, a data object identification request included in the

first message. The system still further includes a query director component for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. The system additionally includes an identification response builder component for, sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. The system also includes a processor, wherein at least one of the content manager component, the query handler component, the query director component, and the identification response builder component includes an instruction that is executed by the processor during operation of the system.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Objects and advantages of the present invention will become apparent to those skilled in the art upon reading this description in conjunction with the accompanying drawings, in which like reference numerals have been used to designate like or analogous elements, and in which:

[0014] FIG. **1** is a block diagram illustrating an exemplary hardware device included in and/or otherwise providing an execution environment in which the subject matter may be implemented;

[0015] FIG. **2A** is a flow diagram illustrating a method for processing a data object identification request in a communication according to an aspect of the subject matter described herein;

[0016] FIG. **2B** is a flow diagram illustrating a method for processing a data object identification request in a communication according to an aspect of the subject matter described herein;

[0017] FIG. **3A** is a block diagram illustrating an arrangement of components for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0018] FIG. **3B** is a block diagram illustrating an arrangement of components for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0019] FIG. **4** is a block diagram illustrating an arrangement of components for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0020] FIG. **5** is a network diagram illustrating a system for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0021] FIG. **6A** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0022] FIG. **6B** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0023] FIG. **6C** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0024] FIG. **6D** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0025] FIG. **6E** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0026] FIG. **6F** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0027] FIG. **6G** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0028] FIG. **7** is a message flow diagram illustrating an exemplary data and execution flow for processing a data object identification request in a communication according to an aspect of the subject matter described herein;

[0029] FIG. **8A** illustrates an exemplary portion of a communication between a first execution environment and a second execution environment according to an aspect of the subject matter described herein; and

[0030] FIG. **8B** illustrates another exemplary portion of a communication between a first execution environment and a second execution environment according to an aspect of the subject matter described herein.

DETAILED DESCRIPTION

[0031] One or more aspects of the disclosure are described with reference to the drawings, wherein like reference numerals are generally utilized to refer to like elements throughout, and wherein the various structures are not necessarily drawn to scale. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of one or more aspects of the disclosure. It may be evident, however, to one skilled in the art, that one or more aspects of the disclosure may be practiced with a lesser degree of these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate describing one or more aspects of the disclosure.

[0032] Unless otherwise defined, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this disclosure belongs. Although methods, components, and devices similar or equivalent to those described herein can be used in the practice or testing of the subject matter described herein, suitable methods, components, and devices are described below.

[0033] All publications, patent applications, patents, and other references mentioned herein are incorporated by reference in their entirety. In case of conflict, the present specification, including definitions, will control. In addition, the materials, methods, and examples are illustrative only and not intended to be limiting.

[0034] An exemplary device included in an execution environment that may be configured according to the subject matter is illustrated in FIG. **1**. An "execution environment", as used herein, is an arrangement of hardware and, in some aspects, software that may be further configured to include and/or otherwise host an arrangement of components for performing a method of the subject matter described herein. An execution environment includes and/or is otherwise provided by one or more

devices. The execution environment is said to be the execution environment of the device and/or devices. An execution environment may be and/or may include a virtual execution environment including software components operating in a host execution environment. Exemplary devices included in and/or otherwise providing suitable execution environments for configuring according to the subject matter include personal computers, notebook computers, tablet computers, servers, portable electronic devices, handheld electronic devices, mobile devices, multiprocessor devices, distributed systems, consumer electronic devices, routers, communication servers, and/or any other suitable devices. Those skilled in the art will understand that the components illustrated in FIG. **1** are exemplary and may vary by particular execution environment.

[0035] FIG. **1** illustrates a hardware device **100** included in an execution environment **102**. FIG. **1** illustrates that execution environment **102** includes a processor **104**, such as one or more microprocessors; a physical processor memory **106** including storage locations identified by addresses in a physical memory address space of processor **104**; a persistent secondary storage **108**, such as one or more hard drives and/or flash storage media; an input device adapter **110**, such as a key or keypad hardware, a keyboard adapter, and/or a mouse adapter; an output device adapter **112**, such as a display and/or an audio adapter for presenting information to a user; a network interface component, illustrated by a network interface adapter **114**, for communicating via a network such as a LAN and/or WAN; and a communication mechanism that operatively couples elements **104-114**, illustrated as a bus **116**. Elements **104-114** may be operatively coupled by various means. Bus **116** may comprise any type of bus architecture, including a memory bus, a peripheral bus, a local bus, and/or a switching fabric.

[0036] As used herein a "processor" is an instruction execution machine, apparatus, or device. A processor may include one or more electrical, optical, and/or mechanical components that operate in interpreting and executing program instructions. Exemplary processors include one or more microprocessors, digital signal processors (DSPs), graphics processing units, application-specific integrated circuits (ASICs), optical or photonic processors, and/or field programmable gate arrays (FPGAs). Processor **104** may access machine code instructions and data via one or more memory address

spaces in addition to the physical memory address space. A memory address space includes addresses identifying locations in a processor memory. The addresses in a memory address space are included in defining a processor memory. Processor **104** may have more than one processor memory. Thus, processor **104** may have more than one memory address space. Processor **104** may access a location in a processor memory by processing an address identifying the location. The processed address may be identified by an operand of a machine code instruction and/or may be identified by a register or other portion of processor **104**.

[0037] FIG. **1** illustrates a virtual processor memory **118** spanning at least part of physical processor memory **106** and may span at least part of persistent secondary storage **108**. Virtual memory addresses in a memory address space may be mapped to physical memory addresses identifying locations in physical processor memory **106**. An address space for identifying locations in a virtual processor memory is referred to as a virtual memory address space; its addresses are referred to as virtual memory addresses; and its processor memory is referred to as a virtual processor memory or virtual memory. The term "processor memory" may refer to physical processor memory, such as processor memory **106**, and/or may refer to virtual processor memory, such as virtual processor memory **118**, depending on the context in which the term is used.

[0038] Physical processor memory **106** may include various types of memory technologies. Exemplary memory technologies include static random access memory (SRAM) and/or dynamic RAM (DRAM) including variants such as dual data rate synchronous DRAM (DDR SDRAM), error correcting code synchronous DRAM (ECC SDRAM), RAMBUS DRAM (RDRAM), and/or XDR™ DRAM. Physical processor memory **106** may include volatile memory as illustrated in the previous sentence and/or may include nonvolatile memory such as nonvolatile flash RAM (NVRAM) and/or ROM.

[0039] Persistent secondary storage **108** may include one or more flash memory storage devices, one or more hard disk drives, one or more magnetic disk drives, and/or one or more optical disk drives. Persistent secondary storage may include a removable data storage medium. The drives and their associated tangible computer readable storage media provide volatile and/or nonvolatile storage for computer-readable

instructions, data structures, program components, and other data for execution environment **102**.

[0040] Execution environment **102** may include software components stored in persistent secondary storage **108**, in remote storage accessible via a network, and/or in a processor memory. FIG. **1** illustrates execution environment **102** including an operating system **120**, one or more applications **122**, and other program code and/or data components illustrated by other libraries and subsystems **124**. In an aspect, some or all software components may be stored in locations accessible to processor **104** in a shared memory address space shared by the software components. The software components accessed via the shared memory address space are stored in a shared processor memory defined by the shared memory address space. In another aspect, a first software component may be stored in one or more locations accessed by processor **104** in a first address space and a second software component may be stored in one or more locations accessed by processor **104** in a second address space. The first software component is stored in a first processor memory defined by the first address space and the second software component is stored in a second processor memory defined by the second address space.

[0041] Software components typically include instructions executed by processor **104** in a computing context referred to as a "process". A process may include one or more "threads". A "thread" includes a sequence of instructions executed by processor **104** in a computing sub-context of a process. The terms "thread" and "process" may be used interchangeably herein when a process includes only one thread.

[0042] Execution environment **102** may receive user-provided information via one or more input devices illustrated by an input device **128**. Input device **128** provides input information to other components in execution environment **102** via input device adapter **110**. Execution environment **102** may include an input device adapter for a keyboard, a touch screen, a microphone, a joystick, a television receiver, a video camera, a still camera, a document scanner, a fax, a phone, a modem, a network interface adapter, and/or a pointing device, to name a few exemplary input devices.

[0043] Input device **128** included in execution environment **102** may be included in device **100** as FIG. **1** illustrates or may be external (not shown) to device **100**. Execution

environment **102** may include one or more internal and/or external input devices. External input devices may be connected to device **100** via corresponding communication interfaces such as a serial port, a parallel port, and/or a universal serial bus (USB) port. Input device adapter **110** receives input and provides a representation to bus **116** to be received by processor **104**, physical processor memory **106**, and/or other components included in execution environment **102**.

[0044] An output device **130** in FIG. **1** exemplifies one or more output devices that may be included in and/or that may be external to and operatively coupled to device **100**. For example, output device **130** is illustrated connected to bus **116** via output device adapter **112**. Output device **130** may be a display device. Exemplary display devices include liquid crystal displays (LCDs), light emitting diode (LED) displays, and projectors. Output device **130** presents output of execution environment **102** to one or more users. In some embodiments, an input device may also include an output device. Examples include a phone, a joystick, and/or a touch screen. In addition to various types of display devices, exemplary output devices include printers, speakers, tactile output devices such as motion-producing devices, and other output devices producing sensory information detectable by a user. Sensory information detected by a user is referred herein to as "sensory input" with respect to the user.

[0045] A device included in and/or otherwise providing an execution environment may operate in a networked environment communicating with one or more devices via one or more network interface components. The terms "communication interface component" and "network interface component" are used interchangeably herein. FIG. **1** illustrates network interface adapter (NIA) **114** as a network interface component included in execution environment **102** to operatively couple device **100** to a network. A network interface component includes a network interface hardware (NIH) component and optionally a network interface software (NIS) component.

[0046] Exemplary network interface components include network interface controller components, network interface cards, network interface adapters, and line cards. A node may include one or more network interface components to interoperate with a wired network and/or a wireless network. Exemplary wireless networks include a BLUETOOTH network, a wireless 802.11 network, and/or a wireless telephony network

(e.g., a cellular, PCS, CDMA, and/or GSM network). Exemplary network interface components for wired networks include Ethernet adapters, Token-ring adapters, FDDI adapters, asynchronous transfer mode (ATM) adapters, and modems of various types. Exemplary wired and/or wireless networks include various types of LANs, WANs, and/or personal area networks (PANs). Exemplary networks also include intranets and internets such as the Internet.

[0047] The terms "network node" and "node" in this document both refer to a device having a network interface component for operatively coupling the device to a network. Further, the terms "device" and "node" used herein may refer to one or more devices and nodes, respectively, providing and/or otherwise included in an execution environment unless clearly indicated otherwise.

[0048] The user-detectable outputs of a user interface are generically referred to herein as "user interface elements" or abbreviated as "UI elements". More specifically, visual outputs of a user interface are referred to herein as "visual interface elements". A visual interface element may be a visual output of a graphical user interface (GUI). Exemplary visual interface elements include windows, textboxes, sliders, list boxes, drop-down lists, spinners, various types of menus, toolbars, ribbons, combo boxes, tree views, grid views, navigation tabs, scrollbars, labels, tooltips, text in various fonts, balloons, dialog boxes, and various types of button controls including check boxes and radio buttons. An application interface may include one or more of the elements listed. Those skilled in the art will understand that this list is not exhaustive. The terms "visual representation", "visual output", and "visual interface element" are used interchangeably in this document. Other types of UI elements include audio outputs referred to as "audio interface elements", tactile outputs referred to as "tactile interface elements", and the like.

[0049] A visual output may be presented in a two-dimensional presentation where a location may be defined in a two-dimensional space having a vertical dimension and a horizontal dimension. A location in a horizontal dimension may be referenced according to an X-axis and a location in a vertical dimension may be referenced according to a Y-axis. In another aspect, a visual output may be presented in a three-dimensional presentation where a location may be defined in a three-dimensional space having a

depth dimension in addition to a vertical dimension and a horizontal dimension. A location in a depth dimension may be identified according to a Z-axis. A visual output in a two-dimensional presentation may be presented as if a depth dimension existed allowing the visual output to overlie and/or underlie some or all of another visual output.

[0050] An order of visual outputs in a depth dimension is herein referred to as a "Z-order". The term “Z-value” as used herein refers to a location in a Z-order. A Z-order specifies the front-to-back and/or back-to-front ordering of visual outputs in a presentation space with respect to a Z-axis. In one aspect, a visual output with a higher Z-value than another visual output may be defined to be on top of or closer to the front than the other visual output. In another aspect, a visual output with a lower Z-value than another visual output may be defined to be on top of or closer to the front than the other visual output.

[0051] A “user interface (UI) element handler” component, as the term is used herein, includes a component of configured to send information representing a program entity for presenting a user-detectable representation of the program entity by an output device, such as a display. A “program entity” is an object included in and/or otherwise processed by an application or executable. The user-detectable representation is presented based on the sent information. Information that represents a program entity for presenting a user detectable representation of the program entity by an output device is referred to herein as “presentation information”. Presentation information may include and/or may otherwise identify data in one or more formats. Exemplary formats include image formats such as raw pixel data, JPEG, video formats such as MP4, markup language data such as hypertext markup language (HTML) and other XML-based markup, a bit map, and/or instructions such as those defined by various script languages, byte code, and/or machine code. For example, a web page received by a user agent from a remote application provider may include HTML, ECMAScript, and/or byte code for presenting one or more UI elements included in a user interface of the remote application. Components configured to send information representing one or more program entities for presenting particular types of output by particular types of output devices include visual interface element handler components, audio interface

element handler components, tactile interface element handler components, and the like.

[0052] A representation of a program entity may be stored and/or otherwise maintained in a presentation space. As used in this document, the term “presentation space” refers to a storage region allocated and/or otherwise provided for storing presentation information, which may include audio, visual, tactile, and/or other sensory data for presentation by and/or on an output device. For example, a buffer for storing an image and/or text string may be a presentation space as sensory information for a user. A presentation space may be physically and/or logically contiguous or non-contiguous. A presentation space may have a virtual as well as a physical representation. A presentation space may include a storage location in a processor memory, secondary storage, a memory of an output adapter device, and/or a storage medium of an output device. A screen of a display, for example, is a presentation space.

[0053] As used herein, the terms "program” and "executable” refer to any data representation that may be translated into a set of machine code instructions and may optionally include associated program data. The terms are used interchangeably herein. Program representations other than machine code include object code, byte code, and source code. Object code includes a set of instructions and/or data elements that either are prepared for linking prior to loading or are loaded into an execution environment. When in an execution environment, object code may include references resolved by a linker and/or may include one or more unresolved references. The context in which this term is used will make clear the state of the object code when it is relevant. This definition can include machine code and virtual machine code, such as Java™ byte code. As used herein, the terms “application”, and “service” may be realized in one or more executables and/or in one or more hardware components. The terms are used interchangeably herein.

[0054] As used herein, an “addressable entity” is program or a portion thereof, specifiable in programming language in source code. An addressable entity is addressable in a program component translated for a compatible execution environment from the source code. Examples of addressable entities include variables, constants, functions, subroutines, procedures, modules, methods, classes, objects, code blocks,

and labeled instructions. A code block includes one or more instructions in a given scope specified in a programming language. An addressable entity may include a value. In some places in this document "addressable entity" refers to a value of an addressable entity. In these cases, the context will clearly indicate that the value is being referenced.

[0055] Addressable entities may be written in and/or translated to a number of different programming languages and/or representation languages, respectively. An addressable entity may be specified in and/or translated into source code, object code, machine code, byte code, and/or any intermediate languages for processing by an interpreter, compiler, linker, loader, and/or other analogous tool.

[0056] As used herein, the term "network protocol" refers to a formal set of rules, conventions and data structures that governs how computers and other network devices exchange information over a network. The rules, conventions, and data structures are said to be specified or defined in a specification and/or schema.

[0057] The term "network path" as used herein refers to a sequence of nodes in a network that are communicatively coupled for transmitting data in one or more data units of a specified network protocol between a pair of nodes in the network.

[0058] A "data unit", as the term is used herein, is a data entity specified according to a network protocol for transmitting data contained in and/or otherwise identified by the data entity. A data unit is transmitted between a pair of nodes in a network path to send the data in and/or otherwise identified by the data unit from a source node to a destination node that includes a protocol endpoint of the network protocol. A network protocol explicitly and/or implicitly specifies and/or otherwise identifies a schema that defines one or more of a format for a valid data unit and a vocabulary for content of a valid data unit. One example of a data unit is an Internet Protocol (IP) packet. The Internet Protocol defines a format for an IP packet that defines a header for identifying a destination address that identifies a destination node and a payload portion for including a representation of data to be delivered to the identified destination node. The terms "data unit", "data packet", and "packet" are used interchangeably herein. One or more data units of a first network protocol may transmit a message of second network protocol. For example, one or more data units of the IP protocol may include a TCP

message. In another example, one or more TCP data units may transmit an HTTP message.

[0059] How data is packaged in one more data units for a network protocol may vary as the data traverses a network path from a source node to a destination node. Data may be transmitted in a single data unit between two consecutive nodes in a network path. Additionally, data may be exchanged between a pair of consecutive nodes in several data units each including a portion of the data. Data received in a single data unit by a node in a network path may be split into portions included in several respective data units for transmitting to a next node in the network path. Portions of data received in several data units may be combined into a single data unit for transmitting by a node in a network path. For purposes of describing the subject matter, a data unit in which data is received by a node is referred to as a different data unit than a data unit in which the data is forwarded by the node.

[0060] A "protocol address", as the term is used herein, for a network protocol is an identifier of a protocol endpoint that may be represented in a data unit of the protocol. For example 192.168.1.1 is an IP protocol address represented in a human readable format that may be included in an address portion of an IP header to identify a source and/or a destination IP protocol endpoint. A protocol address differs from a symbolic identifier, defined below, in that a symbolic identifier, with respect to a network protocol, maps to a protocol address. Thus, "www. mynode.com" may be a symbolic identifier for a node in a network when mapped to the protocol address 192.168.1.1. An identifier may be both a symbolic identifier and a protocol address depending on its role with respect to its use for a particular network protocol.

[0061] Since a protocol endpoint is included in a node and is accessible via a network via a network interface, a protocol address identifies a node and identifies a network interface of the node. A network interface may include one or more NICs operatively coupled to a network.

[0062] A node in a pair of nodes in a network path at one end of the sequence of nodes in the network path and/or the other end is referred to herein as a "path end node". Note that a node may have two NICs with one NIC at each end of a network path. A network path may be included as a portion of another network path that

communicatively couples a same pair of nodes. Data may be transmitted via the sequence of nodes in a network path between path end nodes communicatively coupled via the network path. Data may be transmitted in one or both directions depending on an ordering of the nodes in the sequence.

[0063] The term "hop" as used herein refers to a pair of consecutive nodes in a network path for transmitting, via a network protocol, data sent from a source node to a destination node. A "hop path" is thus a sequence of hops in a network that respectively include a sequence of pairs of consecutive nodes included in transmitting data from a first path end node of the network path to a second path end node of the network path.

[0064] The term "path-based protocol address" as used herein refers to a protocol address for a network protocol that includes a plurality of path segment identifiers that identify portions of a specific network path identified by the path-based protocol address. A "node-based protocol address" is a path-based protocol address that includes a plurality of node identifiers that identify a sequence of nodes in a network path. A "network-interface-based protocol address" is a path-based protocol address that includes a plurality of network interface identifiers that identify a sequence of network interfaces in a network path. A "NIC-based protocol address" is a type of network-interface-based protocol address that includes a plurality of identifiers that identify a sequence of network interface components. A "hop-based protocol address" is a type path-based protocol address since a hop is a type of network path.

[0065] Given the above definitions, note that the terms "network path" and "hop" may be defined in terms of network interfaces. A "network path" and a "hop path" include a sequence of network interfaces in a network that are included in transmitting data between a pair of path end nodes in the network. A "hop" refers to at least part of a network path that includes a pair of consecutive network interfaces in a sequence of network interfaces in a network path. A "network path" is thus a sequence of hops in a network that respectively include a sequence of pairs of consecutive network interfaces included in transmitting data from a first path end node of the network path to a second path end node of the network path.

[0066] The term "network topology" as used herein refers to a representation of protocol endpoints and/or nodes in a network, and representations of hops representing

communicative couplings between and/or among the protocol endpoints and/or nodes in the network. A network may have different network topologies with respect to different network protocols. A network topology may represent physical communicative couplings between nodes in the network. A network topology may represent logical couplings between protocol endpoints and/or nodes of a particular network protocol or a particular type of network protocol.

[0067] The domain name system (DNS) of the Internet operates based on an application layer protocol defined by the DNS. The nodes in the DNS are communicatively coupled via the DNS protocol and may be represented by a logical network topology. A DNS system includes nodes connected via the DNS protocol and thus has a network topology defined by nodes that include protocol endpoints of the DNS protocol. In still another example, a token-ring network has a circular topology at the link layer, but may have a star topology at the physical layer.

[0068] As used herein, an "entity-specific address space" refers to an address space defined for a specific entity where the addresses in the address space operate as identifiers in the context of the entity. An address from an entity-specific address space is referred to herein as an "entity-specific address". An address is "entity-specific" in that what it identifies is based on the entity to which it is specific. Another address having the same form and content may identify a different entity when in an address space specific to another entity. Addresses in an entity-specific address space operate as identifiers in the context of an entity to which they are "specific" as defined by the specific association of the address space and the entity. Without knowledge of the entity to which an entity-specific address space is specific, what an address in the entity-specific address space identifies is indeterminate. The terms "entity-specific address" and "entity-specific identifier" are used interchangeably herein. An entity-specific address may identify an entity included in the entity to which the address is specific or may identify an entity external to the entity to which the address is specific. The fact that an address is entity-specific does not define a scope for the address.

[0069] A portion of a network is a type of entity. A type of entity-specific address space described herein is a scope-specific address space. As used herein, a "scope-specific address space", specific to a particular region of a network, is an address space defined

for the particular network region, where an address in the scope-specific protocol address operates as identifier, according to a network protocol, of a protocol endpoint in a node outside of the particular region when processed in the context of a node in the particular region. The region is indicated by the span of an indicated scope. The terms "region" and "zone" are used interchangeably herein. An address from a scope-specific address space is referred to herein as a "scope-specific protocol address". An address is "scope-specific" in that what protocol endpoint it identifies depends on the region to which it is specific. Another address having the exact same form and content may identify a different protocol endpoint when in an address space that is specific to another region. A protocol address in a scope-specific address space serves as an identifier in the context of a node in a region to which the scope-specific address space is "specific" as defined by an association of the address space and the region indicated by the scope. Without knowledge of the particular region to which a scope-specific address space is specific, what a scope-specific protocol address in the scope-specific address space identifies is indeterminate. The terms "scope-specific protocol address" and "scope-specific protocol identifier" are used interchangeably herein. Types of scope-specific address spaces indicating exemplary spans include site-specific, LAN-specific, subnet-specific, city-specific, business-specific, and node-specific.

[0070] For a network protocol, an address in a scope-specific address space serves as an identifier of a protocol endpoint in a node. Data may be received via the protocol endpoint from a network via one or more network interfaces that operatively couple the node to the network. Data may be sent via the protocol endpoint for transmitting over the network via the one or more network interfaces in the node. Since a protocol endpoint of a network protocol is included in a node and is accessible via a network via a network interface, a protocol address identifying the protocol endpoint also identifies the node and identifies a network interface of the node.

[0071] As used herein, a "node-specific address space" is a scope-specific address space defined for a specific node in a network, where the addresses in the node-specific address space operate as identifiers of nodes and/or network interfaces in the network when processed in the context of the specific node. An address from a node-specific address space is referred to herein as a "node-specific address". An address is

"node-specific" in that what it identifies depends on the node to which is defined as specific. Another address having the exact same form and content may identify a different node when in an address space specific to another node. Addresses in a node-specific address space operate as identifiers in the context of a node to which they are "specific" as defined by the specific association of the address space and the node. Without knowledge of the node to which a node-specific address space is specific, addresses in the node-specific address space are indeterminate. The terms "node-specific address" and "node-specific identifier" are used interchangeably herein. A node-specific address space is a type of scope-specific address space.

[0072] The term "node" is defined above. Note that an identifier of a network interface in a network also identifies a node that includes the network interface. Thus, a network interface-specific address is also a node-specific address. Network interfaces in a node may have their own respective network interface-specific address spaces that are also node-specific. The network interface-specific address spaces may be combined to form a node-specific address space and/or may be managed as separate address spaces. The adjectives "node-specific" and "network interface-specific" may be used interchangeably.

[0073] A scope-specific identifier differs from a scoped address as described in "Request for Comments" (RFC) document RFC 4007 by S. Deering, et al, titled "IPv6 Scoped Address Architecture", published by the IETF in December, 2006 and further described in Application No. 11/962,285, by the present inventor, filed on 2007/12/21, entitled "Methods and Systems for Sending Information to a Zone Included in an Internet Network". A scoped address space is shared by nodes in a given scope. While a link-local scoped address is specific to a particular node, a link-local scoped address simply identifies a network interface component local to the particular node. A loop-back internet address is specific to a node as well. Neither link-local scoped addresses nor loop-back addresses identify one node to another. As such, neither serves as a node-specific identifier as defined above.

[0074] A "scoped address" is described by RFC 3513 and RFC 4007 as an identifier that, in a particular region of a network, serves as a protocol address of a network interface and/or a node in the particular region. The extent of the particular region is

referred to as the scope of the region and thus the scope within which the identifier serves as a protocol address. A particular region included within a scope as indicated by its span. A scoped address is a valid protocol address only within a particular region as indicated by the address's indicated scope. Examples of scope indicators include node-scope where identifiers are valid only to a single node in the indicated span, LAN-scope where identifiers are valid for nodes in the span of a particular LAN, and subnet-scope where identifiers are valid only for nodes in a particular subnet. RFC 3513 currently defines support for link-local scope, site-local scope, a global scope. A data unit transmitted with a scoped address should not be delivered to node that does not have a network interface in the span indicated by the scope.

[0075] "Path information" is any information that identifies a network path and/or hop path for data transmitted via one or more specified network protocols. Path information may be identified by identifying network interfaces, NICs, nodes, and/or hops included in a network path. "Address information" is any information that identifies a protocol address that, for a network protocol, identifies a protocol endpoint. Address information may identify a unicast protocol address for a network protocol. In identifying a protocol endpoint, a protocol address identifies a node and a network interface. Those skilled in the art will understand upon reading the descriptions herein that the subject matter disclosed herein is not restricted to the network protocols described and/or their corresponding OSI layers. For ease of illustration, the subject matter is described in terms of protocols that correspond to OSI layer three, also referred to as network layer protocols, in general. Particular descriptions are based on versions of the Internet Protocol (IP). Address information may identify one or more protocol addresses. Exemplary protocol addresses include IP addresses, IPX addresses, DECNet addresses, VINES Internet Protocol addresses, and Datagram Delivery Protocol (DDP) addresses.

[0076] The term "path-based address" is defined above. A "node-based address" is a path-based address where some or all of the address includes node identifiers that identify a sequence of nodes in a network path. A "network-interface-based address" is a path-based address where some or all of the address identifies includes identifiers of a network interfaces in sequence in a network path. A "NIC-based address" is a type of

network-interface-based address that identifies a sequence of network interface components. A "hop-based address" is a path-based address where some or all of the address identifies one or more hops in a network path. The protocol address types defined are not mutually exclusive.

[0077] The term "metric space", as used herein, refers to a set, as defined in mathematics, where a distance between elements of the set is defined according to a metric. Metric spaces defined in Euclidean geometry are well-known examples. Those skilled in the art of metric spaces, such as Euclidian spaces, will appreciate that a one-to-one mapping may be determined and/or otherwise identified for mapping addresses from a first coordinate space having a first origin for a metric space to addresses from a second coordinate space having a second origin in the metric space. Given a mapping rule between a first scope-specific address space and a second scope-specific address space and a mapping between the second scope-specific address space and a third scope-specific address space based on a third coordinate space identifying a third origin in the metric space, a mapping from the first coordinate space to the third coordinate space may be determined. A mapping between coordinate spaces for a metric space may be included a coordinate shift and/or a rotation, for example. The mapping may be pre-specified and accessible to the nodes in one or both address spaces. Mapping between locations in a number of different metric spaces is well-known in mathematics. For example, a top half of the surface of sphere may be mapped to a plane. Some will further appreciate that some metric spaces may be mapped to other metric spaces. Some of these mappings are one-to-one and onto.

[0078] An "interaction", as the term is used herein, refers to any activity including a user and an object where the object is a source of sensory data detected by the user. In an interaction the user directs attention to the object. An interaction may also include the object as a target of input from the user. The input from the user may be provided intentionally or unintentionally by the user. For example, a rock being held in the hand of a user is a target of input, both tactile and energy input, from the user. A portable electronic device is a type of object. In another example, a user looking at a portable electronic device is receiving sensory data from the portable electronic device whether the device is presenting an output via an output device or not. The user manipulating an

input component of the portable electronic device exemplifies the device, as an input target, receiving input from the user. Note that the user in providing input is detecting sensory information from the portable electronic device provided that the user directs sufficient attention to be aware of the sensory information and provided that no disabilities prevent the user from processing the sensory information. An interaction may include an input from the user that is detected and/or otherwise sensed by the device. An interaction may include sensory information that is detected by a user included in the interaction that is presented by an output device included in the interaction.

[0079] As used herein "interaction information" refers to any information that identifies an interaction and/or otherwise provides data about an interaction between a user and an object, such as a portable electronic device. Exemplary interaction information may identify a user input for the object, a user-detectable output presented by an output device of the object, a user-detectable attribute of the object, an operation performed by the object in response to a user, an operation performed by the object to present and/or otherwise produce a user-detectable output, and/or a measure of interaction. The term "operational component" of a device, as used herein, refers to a component included in performing an operation by the device.

[0080] Interaction information for one object may include and/or otherwise identify interaction information for another object. For example, a motion detector may detect user's head turn in the direction of a display of a portable electronic device. Interaction information identifying the user's head is facing the display may be received and/or used as interaction information for the portable electronic device indicating the user is receiving visual input from the display. The interaction information may serve to indicate a lack of user interaction with one or more other objects in directions from the user different than the detected direction, such as a person approaching the user from behind the user. Thus the interaction information may serve as interaction information for one or more different objects.

[0081] The term "attention information" as used herein refers to information that identifies an attention output and/or that includes an indication to present an attention output. Attention information may identify and/or may include presentation information

that includes a representation of an attention output, in one aspect. In another aspect, attention output may include a request and/or one or more instructions for processing by a processor to present an attention output. The aspects described serve merely as examples based on the definition of attention information, and do not provide an exhaustive list of suitable forms and content of attention information.

[0082] As used herein the term "attention criterion" refers to a criterion that when met is defined as indicating that interaction between a user and an object is or maybe inadequate at a particular time and/or during a particular time period. In other words, the user is not directing adequate attention to the object.

[0083] As used herein, the term "tag" refers to a character string, which may include one or more words, which may be associated with a resource to create an association between the resource and another resource also tagged with the tag. Tags are often used in sharing media, social bookmarking, social news and blog entries to help users search for associated content. In some contexts, the term "tagging" as used herein refers to the process of associating a tag with a resource that can be tagged. As used herein, the term "tagged association" refers to an association that identifies a tag and a resource that is tagged with the tag. A resource is said to be "tagged" with a tag when a tagging process, also referred to as "tagging", has successfully created a tagged association. In other contexts, a "tagging" refers to a tagged association.

[0084] The term "matched tag", as used herein, refers to an association that identifies a tag and a tagging criterion, where the tagging criterion is based on data other than the tag. The tagging criterion may be based additionally on the tag. The term "tagging association", as used herein, refers to a matched tag represented in a memory, either persistent and/or volatile. For example, a tagging association may be realized as a record that includes and/or identifies a tag and a tagging criterion. The terms "matched tag" and "tagging association" are used interchangeably. When the term "matched tag" refers to something other than its tagging association, it will be made clear. For example, a reference to matched tag presented in a user interface clearly identifies a UI element that identifies and/or otherwise represents and/or otherwise identifies a matched tag either realized or to be realized in a tagging association. A tagged association, defined above, may identify a matched tag and a resource that is tagged

with the tag identified by the matched tag. Such an association is also referred to herein as a "matched tag association". A resource may be tagged with a matched tag when a tagging criterion identified by the matched tag is met for the resource.

[0085] As user herein, the term "vocabulary" refers to a set of valid values that can be assigned to and/or included in a data element. With respect to a tag, a vocabulary defines valid tags. A vocabulary may be specified by one or more rules and/or by identifying one or more valid values directly and/or indirectly.

[0086] As used herein, the term" dictionary" refers a vocabulary wherein one or more terms in the vocabulary is assigned a definition. A "definition" as the term is used herein refers to information that identifies semantic information about a data value, such as a t tag. Semantic information may include one or more of a textual description of a meaning of the term, an audio description, a visual description, and information identifying a use or context for the term. A use and/or context may be identified, for example, by identifying a part of speech, a dialect or language, an antonym, a synonym, and/or an example usage of the defined data value.

[0087] As used herein, any reference to an entity "in" an association is equivalent to describing the object as "identified" by the association, unless explicitly indicated otherwise.

[0088] As used herein, the term "communication" refers to information including a message sent and/or for sending via a network between communicants. A message may include text data, audio data, and/or image data. The term "communicant" as used herein refers to a user included in a communication as a sender and/or a receiver of the information. A communicant is represented by a "communications agent" configured to operate in an execution environment to send data to and/or receive data from another communications agent, on behalf of the represented communicant, according to a communications protocol via network. A communications protocol defines and/or otherwise identifies an address space including communications addresses for delivering data sent in a communication from one communications agent to another. A communications protocol is a type of network protocol.

[0089] The term "communicant alias" as used herein refers to an identifier of a communicant in a communication where the communicant alias is not a

communications address included in an address space of a communications protocol for sending and/or receiving data in the communication.

[0090] The term "attachment" as used herein refers to a portion of a communication that includes data from one communicant to another other than data in the message portion. A resource sent as an attachment is data that is typically not presented "inline" or in a message included in a message portion of a communication. Email attachments are perhaps the most widely known attachments included in communications. An email attachment is a file or other resource sent along with an email in a portion of the email separate from a message portion. A communication may include one or more resources as one or more attachments.

[0091] The terms "contactor" and "contactee" identify roles in a communication. A "contactor" provides information for identifying a "contactee" in a communication. A contactee may be included in a communication by a contactor and/or the contactor's communications agent. A contactor and/or a contactee are roles of a user or communicant of a communications agent and may played by a living being, a node, a component, and/or an application. Both a contactor and a contactee are communicants in a communication.

[0092] As used here, the term "tagging- attribute association" refers to an association that identifies one or more tagging criterion and a resource attribute that the one or more tagging criterion are each based on. That is, whether a tagging criterion identified in a tagging-attribute association is met for a specified resource depends on an attribute of the resource identified by the tagging-attribute association. The term attribute, as used herein, includes the resource, as a resource is considered to be an attribute of itself and/or otherwise to include at least some of its attributes.

[0093] As used herein, the term "resource scope" refers a bounded space in which a resource may be identified. The space may be any space in which a resource may be locate and/or otherwise identified. A bounded may be may include one or more of an identifier space such as file system identifier space and/or a URI address space, a storage space, a geospatial space, a demographic space, and a data type – to name a few examples. space, defines a set of resources from which resources may be identified. For example, the files within a specified folder and its subfolders in a file

system define and/or otherwise specify a resource scope. All user accounts that have an administrator security role define another resource scope. The present disclosure describes a resource scope with respect to resources tagged with a specified tag that makeup a set, referred to herein as a "tagged resource set". A tagged resource set may include a resource tagged with a tag identified by a matched tag.

[0094] In specifying a resource scope, one or more criterion may be specified. The terms "criterion" and "constraint" are used interchangeably herein, with respect to a resource scope. A criterion included in specifying a resource scope is referred to herein as a "scope criterion". A scope criterion may be expressed via any suitable manner, such as a file name, a wildcard expression, a regular expression, a data store location, a data base query, geospatial coordinates, a time, a date, a duration, a relationship to a user, and/or an address – too name a few examples. With respect to a tag, a scope criterion included in defining a resource scope differs from a tagging criterion for the tag in that the scope criterion included in defining a resource scope need not be met in order for a resource to be tagged with the tag. Further a resource scope may be used for purposes other than processing related to a tagged resource.

[0095] One or more scope criterion that together specify and/or otherwise define a resource scope is/are referred to herein as a "scope descriptor". When all the criteria in a scope descriptor are met for a resource, the resource is in the resource scope defined by the scope descriptor. The one or more separate criterion in a scope descriptor may be evaluated in any order suitable or desirable in any given embodiment. In an aspect described below, one scope criterion, that may be included in a scope descriptor, may be specified to determine and/or otherwise identify when the scope descriptor is "active". A scope criterion that indicates whether a scope descriptor is active is referred to herein as a "scope condition". In a further aspect a scope descriptor may include no other criterion indicating the resource scope includes all resources accessible to an application, subsystem, execution environment, and/or group of any of the preceding that are processing and/or otherwise making use of the scope descriptor. Those skilled in the art will understand that whether a scope criterion in a scope descriptor is called a scope condition or not depends on the role played by the scope criterion. The subject

matter may be described without referring to a scope criterion in the role of a scope condition without limiting the scope of subject matter.

[0096] Although, those skilled in the art will know that a scope criterion may be defined as "met" when it is true or may be defined as "met" when it is not true; for purposes of the present disclosure, A scope criterion is said to be "met" for a resource when it indicates the resource is in the resource scope defined by the scope descriptor, and is to be "not met" otherwise. A scope criterion in the role of a scope condition in a scope descriptor is said to be "met" when it indicates the scope descriptor is active, as is said to be "not met" otherwise. In an aspect, a scope descriptor may not include a scope condition or may include a scope condition that is always met. Such a scope descriptor is active by definition. Whether a criterion, in a scope descriptor is met or is not met, may depend on an evaluation of the scope criterion that is based on an identifiable entity. For a scope condition, such an entity is referred to herein as a "condition entity". More generally, for a scope criterion, such an entity is referred to herein as a "scoping entity". Exemplary entities include a user, a document, a measure of time, a location, and the like.

[0097] An association between a tag and scope descriptor is referred to herein as a "scoped tag". A scoped tag is also referred to herein as a match set. As defined above, a tagging criterion may be used to determine whether to tag a particular resource with a particular tag. The tagging criterion is associated with the tag by a matched tag. A scope descriptor identifies resource scope that includes a set of resources. Scope descriptors are necessarily associated with resources.

[0098] A "tagging context", as the term is used herein, identifies and/or otherwise includes a set of tagged associations or taggings. The taggings may identify one or more tags and one or more resources. A "tagging context specification", as the term is used herein, is data that defines and/or otherwise specifies a tagging context. A tagging context specification may be stored in a memory location. The memory location may be contiguous or not and may be included in one or more data storage media, which may be volatile or not. A tagging context specification may include and/or otherwise identify any suitable criterion, referred to in general as "context criterion", for determining whether a tagging is in a tagging context defined by the tagging context specification.

For example, a tag contact specification may identify one or more scope descriptors, one or more tags some or all of which may be and/or may be identified by matched tags, and a context condition; all of which are context criterion by definition.

[0099] A "context condition", as the term is used herein, refers to a context criterion that when met indicates that a tagging context is active for one or more taggings. When a context condition is not met, it is said to be inactive or not active for one or more taggings. For example, a context condition in a tagging context specification may be met for a first user. Taggings in the tagging context are "active" for the first user. For a second user, the context condition may not be met. The taggings are not active for the second user. A context condition, or more generally a context criterion, may be based on any suitable data for any suitable entity. A tagging context may be active for a task and not for another and/or a tagging context may be active for one location and not for another, to name a couple of other examples. When a context condition indicates that an associated tagging context is active for a tag, the context condition is herein said to be "met", otherwise the context condition is not met.

[0100] FIG. **3A** illustrates an arrangement of components in a system that operates in an execution environment, such as execution environment **102** in FIG. **1**. The arrangement of components in the system operates to perform the method illustrated in FIG. **2A**. The system illustrated includes a query handler component **302**, a identification request constructor (IRC) component **304**, and a result handler component **306**. A suitable execution environment includes a processor, such as processor **104**, to process an instruction in at least one of a query handler component, a identification request constructor component, and a result handler component. FIG. **3B** illustrates an arrangement of components in a system that operates to perform the method illustrated in FIG. **2B**. The system illustrated includes a content manager component **309**, a query handler component **302**, a query director component **312**, and a identification response builder (IRB) component **314**. A suitable execution environment includes a processor, such as processor **104**, to process an instruction in at least one of a content manager component, a query handler component, a query director component, and a identification response builder (IRB) component.

[0100] Some components, illustrated in the drawings are identified by numbers with an alphanumeric suffix. A component may be referred to generically in the singular or the plural by dropping a suffix of a portion thereof of the component's identifier. For example, window UI elements, such as a window UI element **602a** in FIG. **6A**, a window UI element **602b** in FIG. **6B**, and their adaptations and analogs; are referred to herein generically as a window UI element **602** or window UI elements **602** when describing more than one. Other components identified with an alphanumeric suffix may be referred to generically or as a group in a similar manner.

[0101] The arrangement components illustrated in FIG. **4** may be adapted to operate in a number of execution environments to perform the method illustrated in FIG. **2A** and/or the method illustrated in FIG. **2B**. FIG. **4** is a block diagram illustrating the components of FIG. **3A**, FIG. **3B**, and/or analogs of the components of FIG. **3A** and FIG. **3B**, respectively adapted to operate in an execution environment **401** that includes and/or otherwise is provided by one or more nodes. FIG. **1** illustrates key components of an exemplary device that may at least partially provide and/or otherwise be included in an execution environment. The components illustrated in FIG. **4** may be included in or otherwise combined with the components of FIG. **1** to create a variety of arrangements of components according to the subject matter described herein.

[0102] As stated, the various adaptations of the arrangement in FIG. **3A** as well as the various adaptations of the arrangement in FIG. **3B** illustrated and described herein are not exhaustive. For example, those skilled in the art will see based on the description herein that arrangements of components for performing the method illustrated in FIG. **2A** and the method illustrated in FIG. **2B** may each be distributed across more than one node and/or execution environment. For example, such an arrangements may operate at least partially in a browser in a one node and at least partially in a server in another node interoperating via a network.

[0103] FIG. **5** illustrates a first node **502** and an execution environment **401** of second node **504** as exemplary devices that each may be included in and/or otherwise adapted for providing an instance, adaptation, and/or analog of execution environment **401** in FIG. **4**. As illustrated in FIG. **5**, first node **502** and an execution environment **401** of second node **504** are operatively coupled to a network **506** via respective network

interface components enabling first node **502** and execution environment **401** of second node **504** to exchange data in a communication on behalf of communicants represented by the respective nodes.

[0104] FIG. **4** illustrates communications agent **403** including an adaptation of the arrangement of components in FIG. **3A** and an adaptation of the arrangement of components in FIG. **3B**. A first communications agent **403** may operate, in first node **502**, on behalf of a first communicant to communicate with a second communicant represented by a second communications agent **403** operating in execution environment **401** of second node **504**.

[0105] Execution environment **401** is illustrated in FIG. **4** hosting communications agent **403**. Components in FIG. **4** are referenced for illustrative purposes in describing communications agents operating in an execution environment of first node **502** and an execution environment of execution environment **401** of second node **504**. Exemplary communications agents include email clients, phone clients including Voice over Internet Protocol (VoIP) clients, instant messaging clients, short message service (SMS) clients, multimedia message service (MMS clients), multi-media communications clients including video phone clients, and other data transfer agents.

[0106] Communications agents **403** in FIG. **4**, respectively operating in first node **502** and in execution environment **401** of second node **504** in FIG. **5** may interoperate via respective network stacks **405**. Communications agents **403** may communicate via one or more communications protocols. FIG. **4** illustrates communications protocol component **407** exemplifying a subsystem for exchanging data via network **506** according to one or more communications protocols, such as simple mail transfer protocol (SMTP), an instant messaging protocol, and/or a real-time voice and/or video protocol. A communication between communications agents **403** in first node **502** and execution environment **401** of second node **504** may include more than one type of data and may use one or more communications protocols in exchanging one or more types of data via network **506**.

[0107] Instances, adaptations, and/or analogs of communications agent **403** in FIG. **4**, in first node **502** and in execution environment **401** of second node **504**, respectively, may communicate via discrete messages, a request/reply protocol, a data streaming

protocol, a session and/or connection-oriented protocol, a connectionless protocol, a real-time communications protocol, an asynchronous communication, a store and forward communications protocol, a reliable delivery communications protocol, a best-effort delivery communications protocol, and/or a secure protocol, to name a few communications options.

[0108] FIG. **4** illustrates communications agent **403** including a content manager component **409**. Content manager component **409** may interoperate with communications protocol layer component **407** and/or network stack **405** to receive data in one or more communications via network **506** with another communications agent in another node. Content manager component **409** is operatively coupled, via com-in component **411**, to communications protocol component **407** for receiving the data from the other node.

[0109] Data received in a communication may include one or more resources and/or content types. Exemplary content types include plain text, markup such as hypertext markup language (HTML), audio data, image data, and/or executable data. Executable data may include script instruction(s), byte code, and/or machine code. In FIG. **4**, communications agent **403** includes one or more content handler components **413** to process data received according to its content type. A data type may be identified by a MIME type identifier. Exemplary content handler components **413** include a text/html content handler component for processing HTML representations; an application/xmpp-xml content handler component for processing extensible messaging and presence protocol (XMPP) streams including presence tuples, instant messages, and audio content handlers including and/or configured to retrieve suitable codices; one or more video content handler components for processing video representations of various types; and still image data content handler components for processing various image data representations.

[0110] Content handler component(s) **413** process received data representations and may provide transformed data from the representations to one or more user interface element handler components **415**. One or more user interface element handler components **415** are illustrated in a presentation controller component **417** in FIG. **4**. Presentation controller component **417** may manage visual, audio, and other types of

output for its including application as well as receive and route detected user and other inputs to components and extensions of its including application, communications agent **409**. A user interface element handler component **415** may be adapted to operate at least partially in a content handler component **413** such as a text/html content handler component and/or a script content handler component. Additionally or alternatively, a user interface element handler component in an execution environment **401** may be received in a communication. For example, a communication, such as an email, may include an HTML content type portion and a script content type portion.

[0111] FIG. **6A-G** illustrates various communications agent windows **602** presentable in a presentation space of a display device, such as output device **130** in FIG. **1**. In FIG. **6a**, a communications agent window **602a** includes a contactor user interface (UI) element **604a** for presenting an identifier of a communicant in the role of a contactor in a communication represented by the communications agent window **602a**. A communications agent window **602a** also includes a contactee UI element **606** for presenting one or more contactee identifier(s) identifying one or more communicants in the role of contactee(s) included in the communication. A presentation space **608** is provided by each communications agent window **602a** for presenting a user message UI element **610** for presenting a message addressed to one or more contactees identified in a contactee UI element **606a**. A presentation space **608a** may also be provided for presenting one or more UI controls for exchanging data in and/or otherwise managing a communication. A send UI element **612a** illustrates an exemplary UI element that may correspond to user input to send data in a communication to one or more identified contactees.

[0112] Data to send in a communication to a communications agent in execution environment **401** of second node **504** may be received by one or more content handler component(s) **413** operating in first node **502** to transform the data into one or more data representations suitable for transmitting in the communication and/or suitable for processing by the communications agent in execution environment **401** of second node **504**. The one or more data representations may be provided to content manager component **409** for sending in the communication to execution environment **401** of second node **504**. Content manager component **409** may package the one or more data

representations in a message formatted according to a communications protocol of the communications agent. Communications protocol component **407** may send the data according to the specification(s) of the communications protocol. Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in the communication to a communications agent via a network.

[0113] Content manager component **409** operating in execution environment **401** included in and/or otherwise provided by first node **502** may provide the packaged, encoded, and/or transformed data to communications protocol component **407** via a com-out component **419**. Com-out component **419** as described above operatively couples communications agent **409** to communications protocol component **407** according to an interface provided by communications protocol component **407** for sending data in a communication according to a communications protocol. Communications protocol component **407** may further package and/or otherwise transform the data to send via network stack **405** for delivery via network **506** to execution environment **401** of second node **504**.

[0114] As described above, a communications protocol may operate via one or more nodes in a network in a network path including a first node and an execution environment **401** of second node **504**. Exemplary path nodes include mail relay nodes, phone switch nodes, and proxy nodes such as instant messaging proxies for communicating through firewalls. As indicated, path node **508** illustrates such a node.

[0115] Sending data in a communication requires identifying a contactee. A contactee may be represented by a communications address in an address space of a communications protocol. In one aspect, information identifying a communications address may be received from a user/communicant of a communications agent in a user node. In FIG. **4**, presentation controller **417** and/or a UI element handler **415**, presenting and/or managing interaction with contactor UI element **604** in FIG. **6A** for first node **502**, may receive a contactor alias in response to a user input corresponding to UI element handler **604**. The user/communicant of first node **502** may enter a contactor alias, such as “Dad”, via a keyboard and/or select a predefined communicant

alias presented in a selection UI control element via a UI element handler component **415**. The user input may be detected by input driver **421**. Corresponding input information may be routed to presentation controller **417** by GUI subsystem **423**. GUI subsystem **423** may send presentation information to a display device via a graphics subsystem **425**. Communications agent **403** may identify a communication address associated with the contactor alias, "Dad". Other communications address(es), such as for one or more contactees, may be received similarly and/or in any suitable manner.

[0116] Data may be sent in a communication according to a form or type of the communication and/or other attribute of the communication such as a security attribute, the amount of data to be set, a priority setting, a task setting, and the like. Some forms of communication do not require a session and/or connection between a first node and an execution environment **401** of second node **504** in order to send data between the two nodes, while others do. An email and/or instant message may use a store and forward model of delivery.

[0117] Data may be sent in a communication in response to a communicant input. A contactor may provide an input corresponding to send UI element **612a** in FIG. **6A**. The input may be received by presentation controller component **417** and/or one or more UI element handlers **415** corresponding to send UI element **612a**. In response to detecting the input, presentation controller component **417** may provide data to be sent in the communication to one or more content handler components **413** according to the content type(s) of the data to be sent.

[0118] The one or more content handler components **413** may encode, format, and/or otherwise transform the data for sending in a message, such as an email message. The one or more content handler components **413** may provide data to be sent to content manager **409**, instructing content manager component **409** to send the data in the communication for delivery to a communications agent in execution environment **401** of second node **504**. Content manager component **409** interoperating with com-out component **419** may further format and/or transform the data for sending in the communication according to communications protocol, for example according to an email communications protocol, by communications protocol component **407**. Communications protocol component **407** may send the communication for delivery to

the communications agent in execution environment **401** of second node **504** via network **506**.

[0119] Path node **508** may relay data sent in the communication between first node **502** and execution environment **401** of second node **504**. Path node **508** may determine a next node and/or a network interface in a network path communicatively coupling first node **502** and execution environment **401** of second node **504** for exchanging data in a communication between communications agents **403** in the first node **502** and in the execution environment **401** of second node **504**.

[0120] For session-oriented and/or connection-oriented communication a session and/or connection may be established if a session/connection has not already been established. Data may be sent for delivery to a communications agent identified based on a contactee communications address during session and/or connection setup. For example, for a voice communication a voice communication may be established via a session initiation protocol. Communications protocol component **407** may operate according to the session initiation protocol specifications. Communications protocol component **407** operating in first node **502** may locate a communications agent by communicating with one or more nodes in network **506** according to the session initiation protocol. Communications protocol component **407** may locate a communications agent in execution environment **401** of second node **504**, based on a communications address for the contactee located based on the contactor alias.

[0121] Once a communication session is established, such as a voice session, data may be sent according to the session communications protocol, such as RTP. Data may be sent according to the session initiation protocol in the communication for managing the voice communication session and/or for exchanging text, image, and/or other data outside of the voice session. Path node **508** may be included in session and/or connection setup. Alternatively or additionally, path node **508** may be included in a network path in a session and/or connection.

[0122] With reference to FIG. **2A**, a block **202** illustrates that the method includes receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. Accordingly, a system for processing a data object

identification request in a communication includes means for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. For example, the arrangement in FIG. **3A**, includes query handler component **302** that is operable for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. FIG. **4** illustrates query handler component **402** as an adaptation and/or analog of the query handler component **302** in FIG. **3A**. One or more query handler components **402** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. In FIG. **4**, query handler component **402** is illustrated as a component of communications agent **403**.

[0123] FIG. **7** includes dataflow message **702** that illustrates a data flow internal to execution environment **401** of first node **502** and/or a message received via network **506** by execution environment **401** of first node **502**. Dataflow message **702** illustrates a data object information dataflow to query handler component **402** operating in execution environment **401** of first node **502**. Data object information is identified to and/or otherwise received by query handler component **402**.

[0124] In various aspects, a query handler component **402** may receive data object information from a user via one or more UI element handler components **415**. The data object information may be received, in response to a user input detected by an input device of an execution environment **401** that includes the query handler component **402**. Alternatively or additionally, data object information may be received by a query handler component in a message received via a network. For example, a user may identify data object information to a browser operating in a client node. The browser may send the data object information via a network communicatively coupling the client node to a web service provider node. The web service provider node may include a

query handler component that may operate to receive the data object information identified in the message from the browser.

[0125] FIG. **6A** illustrates an exemplary user interface that may be presented by communications agent **403** operating in execution environment **401** of first node **502**. A UI element handler component **415** may operate to present a user interface element to allow a user to identify data object information. A data object information (DOI) UI element **616a** is illustrated in FIG. **6A** as an example. The same or different UI element handler component **415** may present various other UI elements included in DOI UI element **616a**. DOI UI element **616a** and one or more UI elements it includes may be presented by one or more corresponding UI element handler component(s) **415** that may operate based on a schema that defines valid data object information and/or data object identification requests for including in a communication via a communications agent **403**. The schema defines one or more rules and/or a vocabulary that defines whether data object information and/or a data object identification request is valid. Those skilled in the art will understand that numerous languages and/or schemas for providing data object information and/or data object identification requests currently exist including relatively simple key word based queries to relatively more complicated regular expression languages and data base query languages, such as various versions of structured query language (SQL). New schemas are and will be created, and are, thus, considered within the scope of the subject matter described herein.

[0126] DOI UI element **616a** illustrates a user interface for a keyword based schema. Match textbox UI element **618a** illustrates a textbox allowing a user to enter a keyword expression. Various checkbox UI elements **620a** allow a user to define a scope of a data object identification request by identifying whether a resulting data object identification request is to be applied to folders, files, and/or content of files in a file system and/or other data store of a receiving execution environment. A communications agent **403** may support more than one data object identification request schema and/or corresponding data object information schema. Thus. a communications agent may provide a user interface to receive valid data object information for each respective schema.

[0127] Returning to FIG. **2A**, a block **204** illustrates that the method further includes sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. Accordingly, a system for processing a data object identification request in a communication includes means for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. For example, the arrangement in FIG. **3A**, includes identification request constructor component **304** that is operable for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. FIG. **4** illustrates identification request constructor component **404** as an adaptation and/or analog of identification request constructor component **304** in FIG. **3A**. One or more identification request constructor components **404** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. In FIG. **4**, identification request constructor component **404** is illustrated as a component of communications agent **403**.

[0128] FIG. **7** includes dataflow message **704** that illustrates a data flow internal to execution environment **401** of first node **502** and/or message received via network **506** by execution environment **401** of first node **502**. Dataflow message **704** illustrates a

create data object identification request (CIReq) dataflow to an IRC component **404** operating in execution environment **401** of first node **502**. Data object information may be identified to and/or otherwise provided to IRC component **404**. A query handler component **402**, operating as described with respect to block **202** of FIG. **2A**, may identify and/or otherwise provide data object information to an IRC component **404**.

[0129] A query handler component may be a type of content handler component that operates to process data object information to create a data object identification request that conforms to a schema for creating and/or otherwise constructing a valid data object identification request. The IRC component **404** may construct and/or otherwise create a data object identification request based on a schema that defines and/or otherwise identifies a valid data object identification request and/or a valid data object identification response for a particular type of communication supported by a communications agent. The data object identification response is constructed and/or otherwise created, based on the data object information, by the IRC component **404**. IRC component **404** in execution environment **401** of first node **502** may provide the data object identification request to content manager component **409** to include and/or otherwise identify the data object identification request in a communication with execution environment **401** of second node **504**, such as email and/or instant message. Content manager component **409** may interoperate with other types of content handler components **413** to create and/or otherwise construct a message that includes the valid data object identification request.

[0130] In an aspect, the IRC component **404** may interoperate with the content manager component **409** to create the valid data object identification request according to a specified schema for the message, so that the message is valid according to a communications protocol supported by a communications protocol component **407** that operates to send the message. As described above, the content manager component **409** may operate in an execution environment **401** of first node **502** along with an IRC component **404** to transform data object information into a data object identification request to include along with data for other parts of a communication into one or more representations suitable for transmitting in a communication to another node, such as execution environment **401** of second node **504**. Some or all of the representations

transmitted are suitable for processing by the communications agent in execution environment **401** of second node **504**. The content manager component **409** in the execution environment **401** of first node **502** may package the one or more data representations including a representation of the data object identification request into a message formatted according to the communications protocol.

[0131] In FIG. **8A**, a portion of an email communication **800a** is illustrated formatted as a multipart/mixed content type including data object identification request portion **802a**. A data object identification request portion of a communication may be identified as a data object identification request by its location in the communication and/or by an identifier or markup element, such as a MIME type identifier. A data object identification request may be detected based on content included in the message and/or based on metadata such as content-type header **804a** identifying a MIME type identifier, such as "application/id-request", which may be defined for representing one or more matching criteria in a data object identification request. The "application/id-request" MIME type identifier is exemplary. Other MIME type identifiers exist and/or may be defined to identify a data object identification request in a communication.

[0132] An IRC **404** and/or a content manager component **409** may operate to construct a data object identification request in a communication based on XQuery, regular expression, and/or SQL content – to name a few examples. A content type identifier may be included in a position and/or location that identifies a data object identification request in a communication. The position or location may be absolute or relative. For example, a schema for a communication may define that a data object identification request in a communication is included in the communication at the end of the communication. There may be one or more data object identification requests at the end. In another aspect, a schema for a communication may specify that a portion of a communication following a particular type of message portion is a data object identification request. Other data object identification requests may follow. If no data object identification request is included, the data object identification request portion may include no content or may include an indicator that no data object identification request is included.

[0133] FIG. **8A** illustrates an "application/id-request" MIME type identifier that may be defined to identify a schema for an XML-based language for specifying id-request XML documents. FIG. **8A** illustrates id-request document **806a**. Id-request document **806a**, as illustrated, includes criterion tag elements **808a** corresponding to the form elements in the user interface illustrated in FIG. **6A**. A criterion tag element **808a** identifies a data object matching criterion name, such as "query" indicating that the criterion is a query expression or a portion thereof. A "type" attribute identifies a schema with the identifier "keyword" for the expression. The criterion tag element **808a** identifies a value for a query "Uncle Joe" in a match-expression attribute. Another criterion tag element **808a** specifies a scope query specified in the first criterion tag element. FIG. **8A** illustrates an "and" tag **810a** indicating that all the matching criteria must be met in identifying a data object. An "or" tag (not shown) may be defined by a schema for id-request documents. Other operator elements and operator precedence may be defined by the schema. Grouping elements for managing operator precedence, such as a parenthesis element, may be defined by the schema.

[0134] FIG. **7** includes dataflow message **706** that illustrates a first message, including a data object identification request, sent via network **506** by execution environment **401** of first node **502** to execution environment **401** of second node **504**.

[0135] A data object identification request generated by an IRC component **404**, along with a user message and any other data to include in a communication, may be provided and/or otherwise identified to content manager component **409** for sending in the communication. The content manager component **409** in execution environment **401** of first node **502** may package the data including a representation of the data object identification request into the message formatted according to the communications protocol. Com-out component **419** may provide the data object identification request, the user message, and any other data for sending in the communication in representations suitable for sending by communications protocol component **407** to the communications agent in execution environment **401** of second node **504**. Communications protocol component **407** operating in execution environment **401** of first node **502** may send the data as and/or in a communication, such as an email message, according to the specification(s) of the communications protocol.

Communications protocol component **407** may further package and/or otherwise transform the data to send via network stack **405** for delivery via network **506** to execution environment **401** of second node **504**.

[0136] Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in a communication with the communications agent in execution environment **401** of second node **504**.

[0137] Returning to FIG. **2A**, a block **206** illustrates that the method yet further includes receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. Accordingly, a system for processing a data object identification request in a communication includes means for receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. For example, the arrangement in FIG. **3A**, includes result handler component **306** that is operable for receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. FIG. **4** illustrate result handler components **406** as an adaptations and/or analogs of result handler component **306** in FIG. **3A**. One or more result handler components **406** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is

operable for receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. In FIG. **4**, a result handler component **406** is illustrated as a component of application **403**.

[0138] FIG. **7** includes dataflow message **714** that illustrates a second message, including a data object identification response, received via network **506** by execution environment **401** of first node **502** from execution environment **401** of second node **504**.

[0139] In addition to packaging communication information including a data object identification request into a valid communication, content manager component **409** operating in execution environment **401** of first node **502** may interoperate with communications protocol layer component **407** and/or network stack **405** to receive data in one or more communications via network **506** with another communications agent in another node. A result handler component **406** may receive some or all of the communication in receiving and/or otherwise identifying a data object identification response in the communication. Content manager component **409** in FIG. **4** operatively couples result handler component **406** with network **506**, via com-in component **411**, communications protocol component **407**, and/or network stack **405** to receive the data from the other node. Thus, execution environment **401** of first node **502** may receive a communication, via content manager component **409** and result handler component **406**, that includes a data object identification response that is a response to a data object identification request detected by execution environment **401** of second node **504** in a message received previously from the execution environment **401** of first node **502**.

[0140] Com-in component **411** may provide data received in a communication to content manager component **409**. Content handler component **409** may identify a data object identification response in the communication sent from the execution environment **401** of second node **504**. Content manager component **409** may interoperate with result handler component **406** in identifying the data object identification response. The data object identification response may be identified by content manager component **409** according to a schema for the communication. For

example, a portion of an email message may include a MIME-type identifier that identifies a corresponding part of the email as a data object identification response and/or otherwise identifies the part for routing to a result handler component **406** in communications agent **403** in execution environment **401** of first node **502** for detecting the data object identification response.

[0141] Data sent in second message **714**, by execution environment **401** of second node **504**, may be received by com-in component **411** operating in an instance, adaptation, and/or analog of execution environment **401** including and/or provided by first node **502**. The received communication includes a message addressed to the a communicant represented by communications agent **403** in first node **502** and a data object identification response identifying a data object located and/or otherwise identified based on the data object identification request identified by first message **706**. Content manager component **409** may detect the user message, such as message portion **802b** in second message **714**. Content manager component **409** may detect a data object identification response, such as data object identification response portion **804b** in second message **714**. Message portion **802b** and data object identification response portion **804b** may be provided to suitable content handler components **413** based on the content types of the message portion and the data object identification response portion detected by content manager component **409**. In FIG. **4**, data object identification response portion **804b** may be provided to result handler component **406** as a content handler that matches the MIME type field **806b** illustrated in FIG. **8A**.

[0142] In an aspect, a data object identification response that is a response to a data object identification request sent in a previous message may include information identifying one or more data objects located and/or otherwise identified by the execution environment **401** of second node **504** that processed the data object identification request. For example, a data object identification response may be sent that identifies a URI for each data object located and/or otherwise identified by a first node in response to processing a corresponding data object identification request.

[0143] Returning to FIG. **2A**, a block **208** illustrates that the method yet further includes sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. Accordingly, a

system for processing a data object identification request in a communication includes means for sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. For example, the arrangement in FIG. **3A**, includes access request constructor component **313** that is operable for sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. FIG. **4** illustrates access request constructor component **408** as an adaptation and/or analog of access request constructor component **308** in FIG. **3A**. One or more access request constructor components **408** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. In FIG. **4**, a access request constructor component **408** is illustrated as a component of result handler component **406**.

[0144] FIG. **7** includes dataflow message **716** that illustrates a present message that may be internal to execution environment **401** of first node **502** and/or may be a message sent to a remote device, such as a node including a browser, to send presentation information to present, via an output device, a representation of the data object identification response received in second message **714**. FIG. **7** also includes dataflow message **718** that illustrates a create access request message that may be a data flow internal to execution environment **401** of first node **502** and/or a message received via network **506** by execution environment **401** of first node **502**. In the dataflow message **718** an ARC component **408** operating in execution environment **401** of first node **502** may be invoked to operate based on result information. Result information may be identified to and/or otherwise by ARC component **408** as part of and/or in response to being invoked.

[0145] Result handler component **406** may be a type of content handler component **413** that operates to process data representations and may provide transformed data from the representations to one or more user interface element handler components

415. Result content handler component **406** may operate to process data object identification responses according to a schema defining valid data object identification responses. Content manager component **409** may provide and/or otherwise identify various portions of a communication, such as a data object identification response, to one or more content handler components **413**. The data object identification response may be provided to result handler component **404**. In an aspect, result handler component may interoperate with the presentation controller component **417** in the execution environment **401** of first node **502** to present the data received in the data object identification response and/or other data received in the communication.

[0146] FIG. **6B** illustrates a response window UI element **602b** as an exemplary user interface that may be presented by communications agent **403** operating in execution environment **401** of first node **502** to present data received in a communication from execution environment **401** of second node **504**. FIG. **6B** illustrates a contactor UI element **604b**, a contactee UI element **606b**, a presentation space **608b** of the response window UI element **602b**. Presentation space **608** includes a user message UI element **610b**, analogous to similar UI elements described with respect to FIG. **6A**. The various UI elements may be presented by one or more UI element handler components **415** in the communications agent **402** in execution environment **401** of first node **502**. A reply UI element **622b** and a reply all UI element **624b** illustrate UI elements in presentation space **608b** presented to allow a user of the execution environment **401** of first node **502** to respond and/or otherwise instruct execution environment **401** of first node **502** to perform an operation based on the message received in the communication. FIG. **6B** illustrates a data object identification requested query UI interface element **626b** where the presented term "None" may be defined to indicate no data object identification request was detected by the content manager component **409** in execution environment **401** of first node **502** in the communication received by execution environment **401** of first node **502** from second node **504** via network **506**.

[0147] Response window UI element **602b** in FIG. **6B** includes a data object identification response UI element **628b** for receiving user input to display a representation of one or more data objects located and/or otherwise identified by the

execution environment **401** of second node **504** and identified in the data object identification response. One or more of the data objects is not included in the message received in the communication. In response to an input directed to data object identification response UI element **628b**, a corresponding UI element handler **415** may interoperate with result handler component **406** to present a located data resources dialog UI element **630b** that includes a presentation space **632b** for presenting a user detectable representation of some or all of the data objects identified in the received data object identification response from the execution environment **401** of second node **504**. In an aspect, one or more data objects located and/or otherwise identified by execution environment **401** of second node **504** may be included in a communication as attachments. In a further aspect, located and/or otherwise identified data objects not included in the message may be identified as described with respect to located and/or otherwise identified data objects dialog UI element **630b** and/or in any other suitable manner. In yet another aspect, files and/or other data entities identified in the data object identification response may be illustrated in a manner similar to attachments rather than in a separate UI element as in FIG. **6B**, and accessed from first node **504** via network **506** as described below.

[0148] FIG. **6B** illustrates an access UI element **634b** and an access all UI element **636b** allowing a user to provide user input(s) to instruct communications agent **403** in execution environment **401** of first node **502** to request one or more data objects identified in the data object identification response.

[0149] FIG. **7** includes a dataflow message **720** that illustrates a third message that includes an access request to access a data object identified in the data object identification response received in second message **714**. FIG. **7** illustrates the access request is sent via network **506** by execution environment **401** of first node **502** to execution environment **401** of second node **504**.

[0150] In response to receiving second message **714**, execution environment **401** of first node **502** may present, in a manner described above, data received in the message to a user of execution environment **401** of first node **502** as described with respect to FIG. **6C**. Context menu **646c** provides UI elements to allow the user to instruct execution environment **401** of first node **502** to retrieve a data object identified in a data

object identification response received from execution environment **401** of second node **504**. In response to a user input corresponding to a context menu item included in context menu **646c** a corresponding UI element handler component **415** may invoke ARC component **408** to construct an access request to retrieve one or more data objects identified in the data object identification response. FIG. **6G** illustrates another exemplary user interface that execution environment **401** of first node **502** may present to the user to allow the user to instruct execution environment **401** of first node **502** to construct and send a request. Auto-response window **602g** may be presented in response to a data object identification response received in a communication without a user message. Rather than a UI element for displaying a user message, a results pane UI element **670g** may be presented that identifies data objects located and/or otherwise identified by execution environment **401** of second node **504** that may be retrieved by execution environment **401** of first node **502**. In an aspect, execution environment **401** of second node **504** may process a data object identification request and automatically send a data object identification response without involving the user of execution environment **401** of second node **504**. The data object identification response may be received without a user message as FIG. **6G** illustrates or may be received with a pre-written and/or automatically generated message. UI elements **672g-674g** may be presented by execution environment **401** of first node **502** to allow the user of execution environment **401** of first node **502** to request the retrieval of one or more data objects identified in the data object identification response.

[0151] ARC component **408** may be invoked to construct an access request to retrieve one or more data objects identified in the data object identification response. An access request may be based on a URL, such as an HTTP and/or FTP URL that identifies and locates a data object. Alternatively or additionally, an access request may be sent in a communication supported by a communications agent **403**, in a manner analogous to that illustrated with respect to FIG. 9A. See commonly owned U. S. Patent Application No. 12/833,014 (Docket No 0155) filed on 2010/07/09, entitled "Methods, Systems, and Program Products for Processing a Request for a Resource in a Communication". As described above, content manager component **409** in execution environment **401** of first node **502** may package the one or more data representations including a representation

of an access request into a message in a communication. The message is formatted according to a communications protocol supported by communications agent **403** in execution environment **401** of first node **502**. Communications protocol component **407** operating in execution environment **401** of first node **502** may send the message according to the specification(s) of the communications protocol. Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in a communication.

[0152] Content manager component **409** operating in execution environment **401** included in and/or otherwise provided by execution environment **401** of first node **502** may provide the packaged, encoded, and/or transformed data to communications protocol component **407** via a com-out component **419**. Com-out component **419** as described above operatively couples communications agent **409** to communications protocol component **407** according to an interface provided by communications protocol component **407** for sending data in a communication according to a communications protocol. Communications protocol component **407** may further package and/or otherwise transform the data to send via network stack **405** for delivery via network **506** to execution environment **401** of second node **504**.

[0153] Returning to FIG. **2A**, a block **210** illustrates that the method yet further includes receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. Accordingly, a system for processing a data object identification request in a communication includes means for receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. For example, the arrangement in FIG. 3, includes content director component **310** that is operable for receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. FIG. **4** illustrates content director component **410** as an adaptation and/or analog of content director component **310** in FIG. 3. One or more content director components **410** operate in an execution environment **401**. The system for content director includes one or more processors and logic encoded in one or more tangible media for execution by the one or more

processors that when executed is operable for receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. In FIG. **4**, a content director component **410** is illustrated as a component of communications agent **403**.

[0154] FIG. **7** includes dataflow message **722** that illustrates a fourth message that includes an access response received via network **506** by execution environment **401** of first node **502** from execution environment **401** of second node **504**.

[0155] As described above, content manager component **409** operating in execution environment **401** of first node **502** may interoperate with communications protocol layer component **407** and/or network stack **405** to receive data in a communication, via network **506**, with another communications agent in another node. Content manager component **409** is operatively coupled, via com-in component **411**, to communications protocol component **407** to receive the data from the other node. Thus, execution environment **401** of first node **502** may receive a message in a communication with first node **504** via a com-in component **411**. The message includes an access response to an access request. Content director component **410** may be included in content manager component **409** to detect one or more portions of a received message, such as a portion that is a response to an access request. Content director component **410** may invoke one or more content handler components **413** to process one or more data objects provided in the message according to their type.

[0156] In one aspect, a data object returned in a message in response to access request may be processed as an attachment and/or in a manner analogous to processing of an attachment for a particular communications type.

[0157] With reference to FIG. **2B**, a block **212** illustrates that the method includes receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. Accordingly, a system for processing a data object identification request in a communication includes means for receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user

and operating in a first execution environment. For example, the arrangement in FIG. **3B**, includes content manager component **309** that is operable for receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. FIG. **4** illustrates content manager component **409** as an adaptation and/or analog of content manager component **309** in FIG. **3B**. One or more content manager components **409** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. In FIG. **4**, a content manager component **409** is illustrated as a component of communications agent **403**.

[0158] As described above, FIG. **7** illustrates first message **706**, including a data object identification request, received via network **506** by execution environment **401** of second node **504** from execution environment **401** of first node **502.**

[0159] In addition to packaging representations of data in a structure and/or format for sending a message according to a communications protocol, a content manager component **409** may operate to detect a data object identification request received and/or otherwise identified in a message received in a communication. Content manager component **409** operating in execution environment **401** of second node **504** may receive the message, illustrated by first message **706**, in a communication with execution environment **401** of first node **502**. Com-in component **411** in execution environment **401** of second node **504** may receive the message via communication protocol component **407** and network stack **405**. First message **706** may be delivered to execution environment **401** of second node **504** via network **506** based on a communications address of a communicant represented by communications agent **403** in execution environment **401** of second node **504**. A communications agent may represent more than one communicant, which may be identified by different addresses.

[0160] The first message **706** in FIG. **7** may be received in one or more packets via network **506** by network stack **405** and communications protocol component **407** in an instance of and/or analog of execution environment **401** including and/or otherwise provided by execution environment **401** of second node **504**. The data in the communication may be received by com-in component **411**. Com-in component **411** may provide the data to content manager component **409**. Content manager component **409** may determine one or more content types of the data. The content and/or portions of the content may be provided to one or more content type handler components **413** based on the one or more content types identified by content manager component **409**. For example, FIG. **8A** illustrates message portion **812a** including "text/plain" MIME type identifier **814a** as a content type identifier. Message portion **812a** may be provided to a text/plain content handler **413**. Audio data in a voice communication may be provided to an audio content handler component **413**, and video data in a video communication may be provided to a video content handler component **413**.

[0161] Returning to FIG. **2B**, a block **214** illustrates that the method further includes detecting, by the second communications agent, a data object identification request included in the first message. Accordingly, a system for processing a data object identification request in a communication includes means for detecting, by the second communications agent, a data object identification request included in the first message. For example, the arrangement in FIG. **3B**, includes query handler component **302** that is operable for detecting, by the second communications agent, a data object identification request included in the first message. FIG. **4** illustrates query handler component **402** as an adaptation and/or analog of query handler component **302** in FIG. **3B**. One or more query handler components **402** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for detecting, by the second communications agent, a data object identification request included in the first message. In FIG. **4**, query handler component **402** is illustrated as a component of communications agent **403**.

[0162] FIG. **7** includes a dataflow message **708** that illustrates a detect data object identification request dataflow that may be internal to execution environment **401** of second node **504** and/or may include interoperation with another node via a network. With respect to FIG. **4**, dataflow message **708** may correspond to a data exchange and/or other interoperation between content handler component **409** and query handler component **402** to detect and/or otherwise process the data object identification request received in first message **706** by execution environment **401** of second node **504**. FIG. **7** further includes a dataflow message **710** that illustrates a search message for locating one or more data objects based on the data object identification request. The data object identification request may include and/or otherwise identify a data object matching criterion identified by data object information identified by the communicant represented by execution environment **401** of first node **502**. The one or more data objects may be located and/or otherwise identified by determining a data object or data objects that match the data object matching criterion. The data object(s) may be in a data store accessible to execution environment **401** of second node **504**, such as one or more files in a file system of execution environment **401** of second node **504**. Dataflow message **710** may be internal to execution environment **401** of second node **504** and/or may be a message sent to a remote device, such as database server and/or a network attached storage device providing storage locations for some or all of a file system of execution environment **401** of second node **504**.

[0163] Com-in component **411** in execution environment **401** of second node **504**, as described above, provides the message, received from execution environment **401** of first node **502**, to content manager component **409**. Content manager component **409** is operatively coupled, via com-in component **411**, to communications protocol component **407** for receiving data in communications with other nodes, such as execution environment **401** of first node **502**. Thus, execution environment **401** of second node **504** may receive a message via a com-in component **411** in a communication with execution environment **401** of first node **502**. The message may include a data object identification request based on data object information identified by a user to communications agent **403** in execution environment **401** of first node **502**.

[0164] Data received in a communication, which may for example include a voice session and/or an instant message, may include data of various content types. In FIG. **4**, communications agent **403** in execution environment **401** of second node **504** includes one or more content handler components **413** to process data received according to its content type. Content manager component **409** may detect and/or otherwise identify a portion of the message as the data object identification request sent from execution environment **401** of first node **502**. The data object identification request may be identified by content manager component **409** according to a schema for the message and/or a schema for the data object identification request. For example, a portion of the message may include a MIME-type identifier that identifies the portion as a data object identification request and/or otherwise identifies the portion for routing to a query handler component **402** operating in and/or otherwise on behalf of execution environment **401** of second node **504**. Alternatively or additionally, the data object identification request may have structure and/or content that is valid and identifiable according to a schema for defining and/or otherwise identifying valid data object identification requests. A query handler component, may be provided as a type of content handler component, in an aspect. Alternatively or additionally, a query handler component may interoperate with one or more content handler components to detect a data object identification request in a message received by a content manager component.

[0165] As such, query handler component **402** may operate along with other content handler components **413** to process data representations received in the message **706** from execution environment **401** of first node **502**. Query handler component **402** may also provide transformed data from the representations to one or more user interface element handler components **415**. Query handler component **402** operates to validate and process data object identification requests according to a schema defining valid data object identification requests. Various portions of a message including a data object identification request may be provided to one or more content handler components **413**, including query handler component **402**, to interoperate with presentation controller component **417** in execution environment **401** of second node

504 to present some or all of the received message, including the data object identification request.

[0166] FIG. **6D** illustrates a window, request window UI element **602d**, presenting an exemplary representation of a message received from execution environment **401** of first node **502** in a communication with second node **504**. As in FIG. **6A**, request window UI element **602d** includes a contactor UI element **604d**, a contactee UI element **606d**, a presentation space **608d** of the request window UI element **602d** that includes a user message UI element **610d** including a text portion of the message provided by the communicant represented by execution environment **401** of first node **502**. The various UI elements may be presented by one or more UI element handler components **415** in communications agent **403** operating in execution environment **401** of second node **504**. A reply UI element **648d** and a reply all UI element **648d** are illustrated as exemplary UI elements in presentation space **608d** presented to allow a user of execution environment **401** of second node **504** to respond and/or otherwise perform an operation based on the received message. FIG. **6D** illustrates a data object identification request UI interface element **652d** where the data object identification request identified in the message from execution environment **401** of first node **502** is represented to the communicant represented by the execution environment **401** of second node **504**. In an aspect, the communicant is allowed to authorize the request by providing an input corresponding to a search and reply UI element **654d**. In another aspect, the communicant may be allowed to modify the received data object identification request.

[0167] FIG. **6E**, illustrates a reply window UI element **602e** that, in an aspect, may be presented by communications agent **403** in execution environment **401** of second node **504** in response to an user input corresponding to reply UI element **648d** detected by a corresponding UI element handler component **415**. In an aspect, a user of communications agent **403** operating in in execution environment **401** of second node **504** may provide input authorizing some or all of the processing of a data object identification request. Reply window UI element **602e** in FIG. **6E** includes a search UI element **656e** for receiving user input authorizing and/or otherwise instructing execution environment **401** of second node **504** to locate data objects that match the data object

identification request, illustrated in a data object identification requested UI element **658e**.

[0168] As described herein in an aspect, content manager component **409** may detect content type information to detect a data object identification request in a communication. For example, the message portion illustrated in FIG. **8A** may be received in first message **706**. Content manager component **409** may detect "application/id-request" MIME type identifier in content-type header **804a**. The "application/id-request" MIME type identifier may be defined to identify a data object identification request based on one or more matching criteria for identifying a requested data object. Content manager component **409** may identify data object identification request portion **802a** as including the data object identification request.

[0169] In response to detecting a data object identification request in first message **706**, content manager component **409** may provide some or all of the data object identification request to query handler component **402**. For example, query handler component **402** may be configured to operate according to a schema defining a format and/or a vocabulary for an XML-based language for id-request documents. Content manager component **409** may provide id-request document **806a**, as a data object identification request, to query handler component **402**. Query handler component **402** may operate according to the id-request schema. In an aspect, a query handler component may process more than one data object identification request content type. Alternatively or additionally, execution environment **401** may include multiple query handler components **402** for supporting multiple data object identification request content types.

[0170] Returning to FIG. **2B**, a block **216** illustrates that the method yet further includes generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. Accordingly, a system for processing a data object identification request in a communication includes means for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment,

wherein the data object is identified by processing the data object identification request. For example, the arrangement in FIG. **3B**, includes query director component **312** that is operable for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. FIG. **4** illustrates query director component **412** as an adaptation and/or analog of query director component **312** in FIG. **3B**. One or more query director components **412** operate in an execution environment **401**. The system for query director includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. In FIG. **4**, a query director component **412** is illustrated as a component of application **403**.

[0171] FIG. **7** includes dataflow message **712** that illustrates a create data object identification response data flow to process results returned from search dataflow **710**. In FIG. **4**, dataflow message **712** in FIG. **7** may represent data exchanged between query handler component **402** and query director component **412** to process results returned from a search operation performed based on the data object identification request, as illustrated by dataflow message **710**. Processing the results may include creating and/or otherwise constructing a data object identification response to the data object identification request identified in first message **706**. Dataflow message **712** may be internal to execution environment **401** of second node **504** and/or may include interoperation by execution environment **401** of second node **504** via a network with another node.

[0172] FIG. **6F**, illustrates an identify in response dialog UI element **660f** that, in an aspect, may be presented by communications agent **403** in execution environment **401** of second node **504** in response to processing the data object identification request to locate matching data objects. In an aspect, a user of communications agent **403** operating in execution environment **401** of second node **504** may provide input

corresponding to search UI element **656e** in reply window UI element **602e** in FIG. **6E**. A corresponding UI element handler component **415** may operate to invoke query director component **412** to perform and/or otherwise to provide for performing one or more operations to locate one or more data objects based on the data object identification request. Query director component **412** may interoperate with one or more UI element handlers **415**, directly or indirectly via query handler component **402**, in an aspect, to present the UI element **660f** and its contents. Identify in response dialog UI element **660f** includes results pane UI element **662f** including representations of two data objects located and/or otherwise identified by query director component **412** based on the data object identification request. A user may select one or more data objects to identify in a data object identification response. An include UI element **664f** may be presented, as illustrated, to allow a user to instruct communications agent **403** to identify one or more selected data objects in results pane UI element **662f** in a data object identification response. An include all UI element **666f** is illustrated to allow for user input instructing communications agent **403** to identify all data objects represented in the UI element **660f** in a data object identification response in a communication with execution environment **401** of first node **502**. A cancel UI element **668f** allows a user to end presentation of dialog UI element **660f**. A user may opt to identify no data objects represented in the dialog UI element **660f** and located and/or otherwise identified by query director component **412** based on the data object identification request.

[0173] Data objects located and/or otherwise identified by query director component **412** may be identified to IRB component **414** to construct and/or otherwise create a data object identification response identifying the data objects. As described above, in an aspect, a user may be allowed to select which data objects, if any, are to be identified in a data object identification response from one or more data objects located and/or otherwise identified by query director component **412** based on the data object identification response. In another aspect, communications agent **403** in FIG. **4** may invoke query director component **412** automatically to locate one or more data objects based on a data object matching criterion identified in a data object identification request and in response to detecting a data object identification request in a communication. In another aspect, communications agent **403** may invoke query

director component **412** automatically to locate one or more data objects based on a data object identification request in a communication in response to detecting an indication to present, via a output device, some or all of a user message received in a message in the communication. Query director component **412** may be invoked and/or may operate based on the data object identification request and/or data object matching criterion identified to locate a data object prior to, during, and/or after other content received in the communication is processed.

[0174] The method illustrated in FIG. **2B** may include additional aspects supported by various adaptations and/or analogs of the arrangement of components in FIG. **4**. As described above, in one aspect a data object may be located and/or otherwise identified automatically in response to detecting the data object identification request. In another aspect, locating a data object may include presenting a locator UI element representing a data object identification request. Search UI element **656e** in FIG. **6E** exemplifies one type of locator UI control. Search UI element **656e** may be presented by presentation controller component **417** as described above. A user input corresponding to a search UI element, such as search UI element **656e**, may be detected by presentation controller component **417**. Presentation controller component **417** may instruct query director component **412** to locate the data object based on a data object identification request represented by the data object identification request UI element **658e**.

[0175] In another aspect, a data object identification request UI element may be presented to receive one or more additional matching criteria from a user. Based on the data object identification request UI element and input from the user, query director component **412** may receive one or more additional data object matching criterion information and/or may receive input to modify a data object matching criterion identified in a received data object identification request. Query director component **412** may locate one or more data objects in response to receiving the additional criterion.

[0176] Returning to FIG. **2B**, a block **218** illustrates that the method yet further includes sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. Accordingly, a system for processing a data object identification request in a communication includes means

for sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. For example, the arrangement in FIG. **3B**, includes identification response builder component **314** that is operable for sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. FIG. **4** illustrates identification response builder component **414** as an adaptation and/or analog of identification response builder component **314** in FIG. **3B**. One or more identification response builder components **414** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. In FIG. **4**, a identification response builder component **414** is illustrated as a component of application **403**.

[0177] As described above, FIG. **7** includes second message dataflow message **714** illustrating second message sent via network **506** to execution environment **401** of first node **502** from execution environment **401** of second node **504**. Second message **714** includes a data object identification response in response to the data object identification request from first node **502** received and/or otherwise identified in first message **706**.

[0178] IRB component **414** may provide the data object identification response to content manager component **409** to include in a message, illustrated by second message **714**. As described above, content manager component **409** may receive and/or otherwise identify various portions of a message, including a data object identification response, from one or more content handler components **413**. The data object identification response may be provided to content manager component **409**. As described above, content manager component **409** in execution environment **401** of second node **504** may package the one or more data representations including a

representation of the data object identification response into the message formatted according to a suitable communications protocol. Communications protocol component **407** operating in execution environment **401** of second node **504** may send the data as a message in a communication with execution environment **401** of first node **502** according to the specification(s) of the communications protocol. Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in the communication to the communications agent.

[0179] FIG. **8B** illustrates an exemplary data object identification response described in detail above. Second message portion **802b** illustrates a user message portion that may be included in second message **714**. Note that a user message portion may be empty or absent in some aspects. Data object identification response portion **804b** illustrates the data object identification response as a list of URIs identifying data objects in a file system of execution environment **401** of second node **504**. User Message portion **802b** and data object identification response portion **804b** may be provided by respective content handler components **413** to content manager component **409**. Content manager component **409** may construct the content as illustrated in FIG. **8B** and/or otherwise provide for encoding, translating, combining, and/or otherwise preparing the content for sending in second message **714**.

[0180] The methods illustrated in FIG. **2A-B** may include additional aspects supported by various adaptations and/or analogs of the arrangement of components in FIG. **3A-B**. In various aspects, a data object identification request may be based on a date, a time, a length of time, a file type, a database record key, content of the data object, a content type identifier, a format rule, a vocabulary, a role of a user, a security attribute, a location in a data store such as a file system, an attribute of an identified data object, a size, a task, a transaction, a state, a user, a group, a requester, a relationship including a requesting user and a responding user, a keyword, a tag, a folder, and/or a path portion of a data object identifier – to name a few examples.

[0181] In another aspect, a data object located and/or otherwise identified in response to receiving a data object identification request must meet match a query identified by

the data object identification request. Alternatively or additionally, a data object identification request may include an instruction and/or input for generating a data object. Locating the data object may include generating the data object. Generating a data object may include creating the data object and/or may include modifying and/or otherwise transforming an existing data object. For example, execution environment **401** of second node **504** may include a template stored in a file system. A data object identification request may identify the template. A query handler component **402** may operate to identify a document or other data object that may be created, is being created, and/or has been created based on the template. In response to an access request, execution environment **401** of second node **504** may return the document.

[0182] As described above, in various aspects, a data object identification request may be detected and/or represented based on various syntaxes, grammars, vocabularies, and/or languages. For example, a data object identification request may be identified and/or represented according to a file system search syntax, a regular expression language, a structured query language (SQL) query, a universal data object identifier schema, an XPATH based language, an XQuery based language, an XML based language, an HTML based language (form-based), and/or a keyword-value pair based language.

[0183] A data object identification request in a communication may be communicated via a network according to a first communications protocol. A user message may be exchanged between communicants in the communication via a second communications protocol. For example, a data object identification request may be sent on behalf of a first communicant by execution environment **401** of first node **502** including a communications agent **403** representing the first communicant. The data object identification request may be included in a communication along with a text message sent to the execution environment **401** of second node **504**. The communicants represented by the execution environment **401** of first node **502** and the execution environment **401** of second node **504** may communicate via one or more exchanges of audio exchanged via, for example, a voice over IP (VoIP) communications protocol.

[0184] Exemplary data objects that may be requested via a data object identification request include a file, a program component, a data base record, video data, audio

data, markup language, binary data, text data, an output of a service. Requested data objects may be pre-existing, volatile, and/or generated in response to the request.

[0185] As described above, receiving data object information may include receiving, via network from a node that received the data object information from the user, a message that identifies the data object information. For example, data object information may be received from a browser by a web server.

[0186] Data object information and/or a data object identification request may be generated, detected, and/or otherwise processed according to a schema that identifies at least one of a rule and a vocabulary that defines a valid data object identification request. Similarly, a data object identification response, an access request, and a response to an access request may be generated, detected, and/or otherwise processed according to a schema. One or more schemas may alone or together define one or more of valid data object information, a data object identification request, data object identification response, an access request, and/or an access response. A schema, as just described, may be identified based on data received and/or otherwise identified by a node hosting a communications agent via the network from another node hosting a communications agent and/or from a path node included in a network path in the network communicatively coupling the node to one or more nodes that respectively host communications agents.

[0187] In a further aspect, multiple alternative schemas may be defined for each of data object information, data object identification requests, data object identification responses, access requests, an/or an access responses processing data object identification requests. A schema may be selected and/or otherwise identified from the multiple schemas by execution environment **401** of first node **502** and the execution environment **401** of second node **504** and/or based on information exchanged by the two nodes. For example, a schema may be selected by the first communicant and/or the second communicant. One may inform the other, via a communications protocol and/or via any other suitable means. The communicants may negotiate which schema to use via any suitable means.

[0188] The communications agents in execution environment **401** of second node **504** and execution environment **401** of first node **502** may select a schema For example, a

schema may be identified and/or otherwise selected based on a communications protocol supported and/or a communications protocol not supported by one or both nodes. That is, in negotiating which communication protocol to use allowing communicants to exchange information, a schema may be identified. A schema may be identified and/or otherwise selected based on an attribute of the data object. A schema for identifying image data may be different than a schema for editable documents. A schema for identifying a data object in a first file system may be different than a schema for identifying a data object in a different file system.

[0189] As described above, data object information and/or a data object identification request may identify a scope that specifies all or some portion of one or more data stores for locating a data object. For example, A communications agent **403** in execution environment **401** of first node **502** may receive user input that identifies a folder in a file system of the execution environment **401** of second node **504** that defines a scope for locating a data object.

[0190] Data object information and/or a data object identification request may be represented in a representation that includes a complete or a portion of at least one of a keyword expression, a regular expression, an XQuery expression, an XPath expression, a file system path expression, and a structured query language statement.

[0191] Further as also described above, a data object identification request, a data object identification response, an access request, an/or an access response may be included in a message according to a schema that identifies the message as valid for a specified communications protocol. Those skilled in the art are aware that schemas for email differ from schemas for instant messages and from streamed audio, for example.

[0192] A data object identification request, a data object identification response, an access request, and/or an access response may be exchanged in a communication in which audio data and/or image data is also exchanged. Multimedia may be a packaged to include one or more data streams which may include text based data streams.

[0193] A data object identification request, a data object identification response, an access request, and/or an access response may be included in an email, an instant message, a multi-media message, a short-message service message, and/or a data stream. The data stream may include audio data, image data, and/or text data.

[0194] A data object identification request, a data object identification response, an access request, an/or an access response may be included in and/or detected in a communication based on a location in the communication and/or based on a marker, such as an identifier in the communication that identifies a location for the data object identification request, the data object identification response, the access request, and/or the access response. The location may be absolute, such as fixed number of bytes or fields from the start or end of a message. Alternatively or additionally, the location may be identified by a location relative to another detectable portion of the communication. FIGs. **8A-B** illustrate the use of MIME type identifiers as markers that identify locations for one or more of a data object identification request, a data object identification response, an access request, an/or an access response in a communication.

[0195] A data object identification request, a data object identification response, an access request, an/or an access response or respective portions thereof may include at least a portion of a keyword expression, a regular expression, expression including a Boolean operator, an expression including a precedence information, and a structured query language statement.

[0196] With respect to the method illustrated in FIG. **2B**, the method as described above may include locating and/or otherwise identifying a data object automatically in response to receiving an access request that matches and/or otherwise identifies the data object. Further, a data object identification response may be created and/or otherwise generated automatically, in response to locating and/or otherwise identifying the data object.

[0197] In an aspect, in response to receiving and/or otherwise detecting a data object identification request, a user interface may be presented based on the data object identification request to prompt a user to process the data object identification request in some manner, such as authorizing the request and/or modifying the request by restricting its scope. In response to detecting user input targeting and/or otherwise corresponding to the presented user interface, one or more data objects may be located and/or otherwise identified based on the data object identification request.

[0198] A user may change a received data object identification request. One or more data objects may be located and/or otherwise identified based on the changed data object identification request.

[0199] In a further aspect, once one or more data objects are located and/or otherwise identified based on a data object identification request, a user interface may be presented to allow a user to identify the one or more data objects. User information may be received that selects one or more data objects to identify in a data object identification response and/or to not identify in the data object identification response. A user, in an aspect, may be allowed to identify a data object not located and/or otherwise identified by the user's node. The data object, based on the user input, is identified in the data object identification response.

[0200] In light of the description provided above, a data object identification response may be generated automatically in response to detecting a corresponding data object identification request by an execution environment **401** of first node **502**. Further, the automatically generated data object identification response, may be sent to the node that sent the corresponding data object identification request, automatically in response to the generating of the data object identification response.

[0201] With respect to the method illustrated in FIG. **2A**, the third message may be sent automatically in response to detecting the data object identification response. In another aspect, a representation of a received data object identification response may be presented, via an output device, to a user. The representation may identify one or more data objects located and/or otherwise identified based on the data object identification request. The representation may be presented automatically, in response to detecting the data object identification response. Further, a user input may be detected that corresponds to and/or otherwise targets the representation. The user input may be processed as an indication to retrieve the data object. As a result an access request may be created or otherwise generated that identifies the data object. The access request may be sent in a message via the network to retrieve the data object. The message may be sent to a node that sent the data object identification response and/or to another node. The third message may be sent by the communications agent in the execution environment **401** of first node **502** to the communications agent in the

execution environment **401** of second node **504** via a communications protocol. The third message sent via the communication protocol may include a communications address, in an address space of the communications protocol, that identifies a user represented by the communications agent in the execution environment **401** of second node **504**. The third message may include a data object identification request, a data object identification response, and/or a data object sent in response to previously receiving an access response by the execution environment **401** of first node **502**. The third message may include an access request that identifies the data object. The access request may include a universal data object identifier (URI) that identifies the data object.

[0202] With respect to the method illustrated in FIG. **2B**, in response to receiving an access request, a data object identified in the request may be sent. The data object may be sent automatically, in response to receiving the access request. The data object may be sent as attachment in a communication. The data object may be sent by a communications agent in the execution environment **401** of second node **504** to a communications agent in the execution environment **401** of first node **502** via a communications protocol. The data object may be sent via a communication that includes a communications address, in an address space of the communications protocol, that identifies the first communicant represented by the execution environment **401** of first node **502**. The communication that includes the data object may include a data object identification request, a data object identification response, and/or an access request sent in response to a data object identification response sent previously by the first node.

[0203] The methods illustrated in FIGs. **2A-B** may yet further include additional aspects supported by various adaptations and/or analogs of the arrangement of components in FIGs. **3A-B**, respectively.

[0204] Performing the method illustrated in FIG. **2A** and/or the method illustrated in FIG. **2B** and/or any of its extension and/or in any of its aspects may include one or more of calling a function or method of an object, sending a message via a network; sending a message via an inter-process communication mechanism such as a pipe, a

semaphore, a shared data area, and/or a queue; and/or receiving a request such as poll and responding to invoke, and sending an asynchronous message.

[0205] Output devices suitable for presenting a representation of a data object and/or a tag include a visual output device, an audio output device, and a tactile output device. One output device may present a data object and another output device may present a tag with which the data object is tagged.

[0206] Any action included in performing the method illustrated in FIG. **2A** as well as any action included in in performing the method illustrated in FIG. **2B** may include sending and/or receiving a message via a network. Further, a message, included in performing any of the subject matter described herein and/or any of its extensions in any of its aspects, may an asynchronous message without a corresponding request.

[0207] To the accomplishment of the foregoing and related ends, the descriptions and annexed drawings set forth certain illustrative aspects and implementations of the disclosure. These are indicative of but a few of the various ways in which one or more aspects of the disclosure may be employed. The other aspects, advantages, and novel features of the disclosure will become apparent from the detailed description included herein when considered in conjunction with the annexed drawings.

[0208] It should be understood that the various components illustrated in the various block diagrams represent logical components that are configured to perform the functionality described herein and may be implemented in software, hardware, or a combination of the two. Moreover, some or all of these logical components may be combined, some may be omitted altogether, and additional components may be added while still achieving the functionality described herein. Thus, the subject matter described herein may be embodied in many different variations, and all such variations are contemplated to be within the scope of what is claimed.

[0209] To facilitate an understanding of the subject matter described above, many aspects are described in terms of sequences of actions that may be performed by elements of a computer system. For example, it will be recognized that the various actions may be performed by specialized circuits or circuitry (e.g., discrete logic gates interconnected to perform a specialized function), by program instructions being executed by one or more processors, or by a combination of both. The description

herein of any sequence of actions is not intended to imply that the specific order described for performing that sequence must be followed.

[0210] Moreover, the methods described herein may be embodied in executable instructions stored in a non-transitory computer readable medium for use by or in connection with an instruction execution machine, system, apparatus, or device, such as a computer-based or processor-containing machine, system, apparatus, or device. As used here, a "non-transitory computer readable medium" may include one or more of any suitable media for storing the executable instructions of a computer program in one or more forms including an electronic, magnetic, optical, and electromagnetic form, such that the instruction execution machine, system, apparatus, or device may read (or fetch) the instructions from the non-transitory computer readable medium and execute the instructions for carrying out the described methods. A non-exhaustive list of conventional exemplary non-transitory computer readable media includes a portable computer diskette; a random access memory (RAM); a read only memory (ROM); an erasable programmable read only memory (EPROM or Flash memory); optical storage devices, including a portable compact disc (CD), a portable digital video disc (DVD), a high definition DVD (HD-DVD.TM.), and a Blu-ray.TM. disc; and the like

[0211] Thus, the subject matter described herein may be embodied in many different forms, and all such forms are contemplated to be within the scope of what is claimed. It will be understood that various details may be changed without departing from the scope of the claimed subject matter. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation, as the scope of protection sought is defined by the claims as set forth hereinafter together with any equivalents.

[0212] Thus, the subject matter described herein may be embodied in many different forms, and all such forms are contemplated to be within the scope of what is claimed. It will be understood that various details may be changed without departing from the scope of the claimed subject matter. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation, as the scope of protection sought is defined by the claims as set forth hereinafter together with any equivalents.

[0213] All methods described herein may be performed in any order unless otherwise indicated herein explicitly or by context. The use of the terms "a" and "an" and "the" and similar referents in the context of the foregoing description and in the context of the following claims are to be construed to include the singular and the plural, unless otherwise indicated herein explicitly or clearly contradicted by context. The foregoing description is not to be interpreted as indicating that any non-claimed element is essential to the practice of the subject matter as claimed.

I CLAIM:

1. A method for processing a data object identification request in a communication, the method comprising:

receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion;

sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user; and

receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message,

wherein performing at least one of the preceding actions comprising the method includes execution of an instruction by a processor.

2. The method of claim 1 wherein the data object identification request identifies a scope specifying at least a portion of the second data store.

3. The method of claim 1 sending the first message comprises including the data object identification request in the first message separate from a user text message included in the first message.

4. The method of claim 1 wherein the data object identification request includes information for the second execution environment for at least one of generating and modifying the data object.

5. The method of claim 1 receiving the second message includes detecting the data object identification response in the second message along with a user text message from the second user.

6. The method of claim 1 wherein the second message is received in a communication that includes an exchange between the first execution environment and the second execution environment of at least one of audio data and video data via at least one of the first communications protocol and a second communications protocol.

7. The method of claim 1 further includes:

presenting, via an output device, a representation of the data object identification response, that identifies the data object to the first user;

detecting, in response to a user input corresponding to the representation of the identified data object, an indication to retrieve the data object;

creating, based on the data object identification response, an access request identifying the data object; and

sending the access request in a third message.

8. The method of claim 1 further includes

sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object; and

receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object.

9. The method of claim 8 wherein the third message is sent automatically in response to detecting the data object identification response.

10. The method of claim 9 wherein the at least a portion of the data object is received by the first execution environment in a fourth message as an attachment.

11. A method for processing a data object identification request in a communication, the method comprising:

receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment;

detecting, by the second communications agent, a data object identification request included in the first message;

generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request; and

sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object,

wherein performing at least one of the preceding actions comprising the method includes execution of an instruction by a processor.

12. The method of claim 11 wherein the data object identification request identifies a scope specifying a portion of the second data store.

13. The method of claim 11 wherein the data object identification request in the first message is received in a communication that includes an exchange between the first execution environment and the second execution environment of at least one of audio data and video data.

14. The method of claim 11 receiving the first message includes detecting the data object identification request in the first message separate from a user text message included in the first message.

15. The method of claim 11 wherein the data object identification request includes information for the second execution environment for at least one of generating and modifying the data object.

16. The method of claim 11 wherein the data object identification response is generated automatically, in response to receiving the data object identification request.

17. The method of claim 11 further includes:

presenting a locator user interface element based on the data object identification request;

receiving data object identification request change information in response to a user input, detected via an input device; and

locating the data object based on the change information.

18. The method of claim 11 further includes:

sending presentation information to present, to the second communicant, a plurality of data objects located based on the data object identification request;

receiving selection information identifying the data object in the presented plurality; and

generating the data object identification response based on receiving the selection information.

19. The method of claim 11 sending the second message includes including the data object identification response in the second message separate from a user text message included in the second message.

20. The method of claim 11 wherein the second message is sent in a communication that includes an exchange between the first execution environment and the second execution environment of at least one of audio data and video data.

21. The method of claim 11 further including:

receiving a third message, via the network from the first execution environment by the second execution environment, to access the data object identified by the data object identification response; and

sending, via the network the data object to the first execution environment by the second execution environment, the resource,.

22. A non-transitory computer readable medium embodying a computer program, executable by a machine, for processing a data object identification request in a communication, the computer program comprising executable instructions for:

receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment;

detecting, by the second communications agent, a data object identification request included in the first message;

generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request; and

sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object.

ABSTRACT

Methods and systems are described for processing a data object identification request in a communication. In an aspect, a first communications agent in a first execution environment receives data object information identifying a data object matching criterion. The first communications agent exchanges a first message with a second communications agent in a second execution environment. The second communications agent detects a data object identification request included in the exchanged first message. In response to detecting the data object identification request, a data object identification response is generated that identifies a data object in a second data store in the second execution environment. The second communications agent exchanges a second message with the first communications agent. The second message includes the data object identification response, wherein the second message does not include the data object.

Electronic Acknowledgement Receipt

EFS ID:	18995566
Application Number:	14274623
International Application Number:	
Confirmation Number:	4156
Title of Invention:	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION
First Named Inventor/Applicant Name:	Robert Paul Morris
Customer Number:	92045
Filer:	Patrick Edgar Caldwell
Filer Authorized By:	
Attorney Docket Number:	DR0149-CIP
Receipt Date:	09-MAY-2014
Filing Date:	
Time Stamp:	21:24:29
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	DR0149-CIP_POA_vF_09-May-2014.pdf	477402 ba16d5b1ac0e843162efd56a96b4be2b8617057c	no	1

Warnings:	
Information:	

2	Application Data Sheet	DR0149-CIP_ADS_vF_09-May-2014.pdf	74419 180e31b25d2313ef4d88858af6b9c85a6629df46	no	6
Warnings:					
Information:					
This is not an USPTO supplied ADS fillable form					
3	Drawings-only black and white line drawings	DR0149-CIP_Drawings_vF_09-May-2014.pdf	163355 508d383955dbcc8957aaae78426d006d3169f249	no	15
Warnings:					
Information:					
4		DR0149-CIP_Application_vF_09-May-2014.pdf	1862708 1a33141270fd00723cfaed6377c1fb89db8360b6	yes	78

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Specification	1	72
	Claims	73	77
	Abstract	78	78

Warnings:	
Information:	
Total Files Size (in bytes):	2577884

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Document code: WFEE

United States Patent and Trademark Office
Sales Receipt for Accounting Date: 10/18/2017

ASAHLE ADJ #00000006 Mailroom Dt: 10/03/2017
Seq No: 5874 Sales Acctg Dt: 10/04/2017 15724223
05 FC : 2081 -200.00 OP

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82A (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	
Filing Date	10/3/2017
First Named Inventor	Robert Paul Morris
Title	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION
Art Unit	
Examiner Name	
Attorney Docket Number	

SIGNATURE of Applicant or Patent Practitioner

Signature	/Patrick Caldwell/	Date (Optional)	
Name	Patrick E. Caldwell	Registration Number	44,580
Title (if Applicant is a juristic entity)			
Applicant Name (if Applicant is a juristic entity)			

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. If more than one applicant, use multiple forms.

☑ *Total of 2 forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 92045

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

	Firm or Individual Name				
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

Cedar Point Partners, LLC

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature	/Robert Paul Morris/	Date (Optional)	
Name	Robert Paul Morris		
Title	Manager - Cedar Point Partners, LLC		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☑ Total of 2 forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Doc Code: TRACK1.REQ
Document Description: TrackOne Request

PTO/AIA/424 (04-14)

CERTIFICATION AND REQUEST FOR PRIORITIZED EXAMINATION UNDER 37 CFR 1.102(e) (Page 1 of 1)

First Named Inventor:	Robert Paul Morris	Nonprovisional Application Number (if known):	
Title of Invention:	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION		

APPLICANT HEREBY CERTIFIES THE FOLLOWING AND REQUESTS PRIORITIZED EXAMINATION FOR THE ABOVE-IDENTIFIED APPLICATION.

1. The processing fee set forth in 37 CFR 1.17(i)(1) and the prioritized examination fee set forth in 37 CFR 1.17(c) have been filed with the request. The publication fee requirement is met because that fee, set forth in 37 CFR 1.18(d), is currently $0. The basic filing fee, search fee, and examination fee are filed with the request or have been already been paid. I understand that any required excess claims fees or application size fee must be paid for the application.

2. I understand that the application may not contain, or be amended to contain, more than four independent claims, more than thirty total claims, or any multiple dependent claims, and that any request for an extension of time will cause an outstanding Track I request to be dismissed.

3. The applicable box is checked below:

I. ☑ Original Application (Track One) - Prioritized Examination under § 1.102(e)(1)

i. (a) The application is an original nonprovisional utility application filed under 35 U.S.C. 111(a). This certification and request is being filed with the utility application via EFS-Web.

---OR---

(b) The application is an original nonprovisional plant application filed under 35 U.S.C. 111(a). This certification and request is being filed with the plant application in paper.

ii. An executed inventor's oath or declaration under 37 CFR 1.63 or 37 CFR 1.64 for each inventor, **or** the application data sheet meeting the conditions specified in 37 CFR 1.53(f)(3)(i) is filed with the application.

II. ☐ Request for Continued Examination - Prioritized Examination under § 1.102(e)(2)

i. A request for continued examination has been filed with, or prior to, this form.
ii. If the application is a utility application, this certification and request is being filed via EFS-Web.
iii. The application is an original nonprovisional utility application filed under 35 U.S.C. 111(a), or is a national stage entry under 35 U.S.C. 371.
iv. This certification and request is being filed prior to the mailing of a first Office action responsive to the request for continued examination.
v. No prior request for continued examination has been granted prioritized examination status under 37 CFR 1.102(e)(2).

Signature /Patrick Caldwell/	Date 10/3/2017
Name (Print/Typed) Patrick E. Caldwell	Practitioner Registration Number 44,580

Note: *This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. Submit multiple forms if more than one signature is required.**

☑ *Total of 1 forms are submitted.

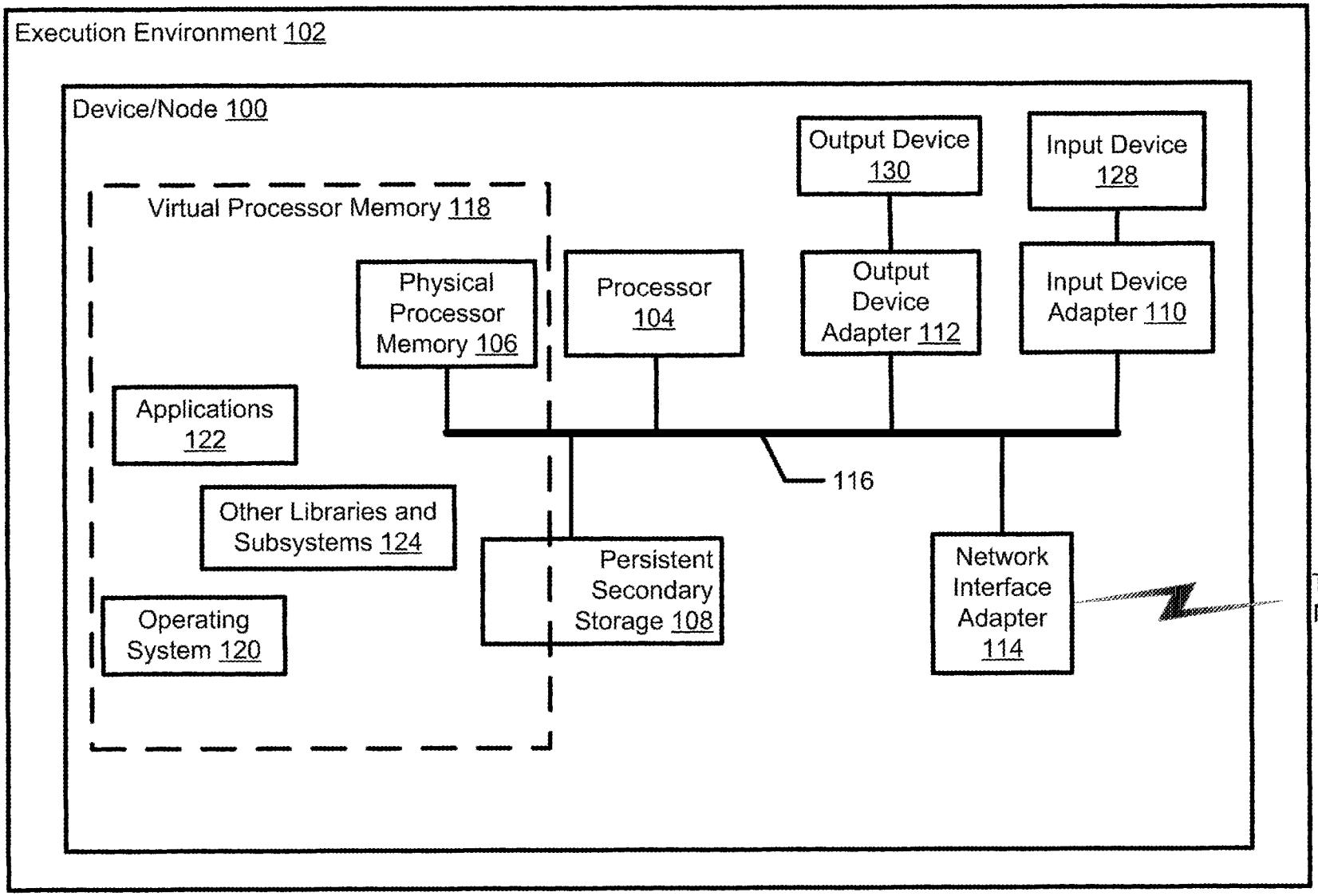

Execution Environment 102
Device/Node 100
Virtual Processor Memory 118
Physical Processor Memory 106
Processor 104
Output Device 130
Input Device 128
Output Device Adapter 112
Input Device Adapter 110
Applications 122
Other Libraries and Subsystems 124
Operating System 120
Persistent Secondary Storage 108
116
Network Interface Adapter 114
To/From Network

FIG. 1

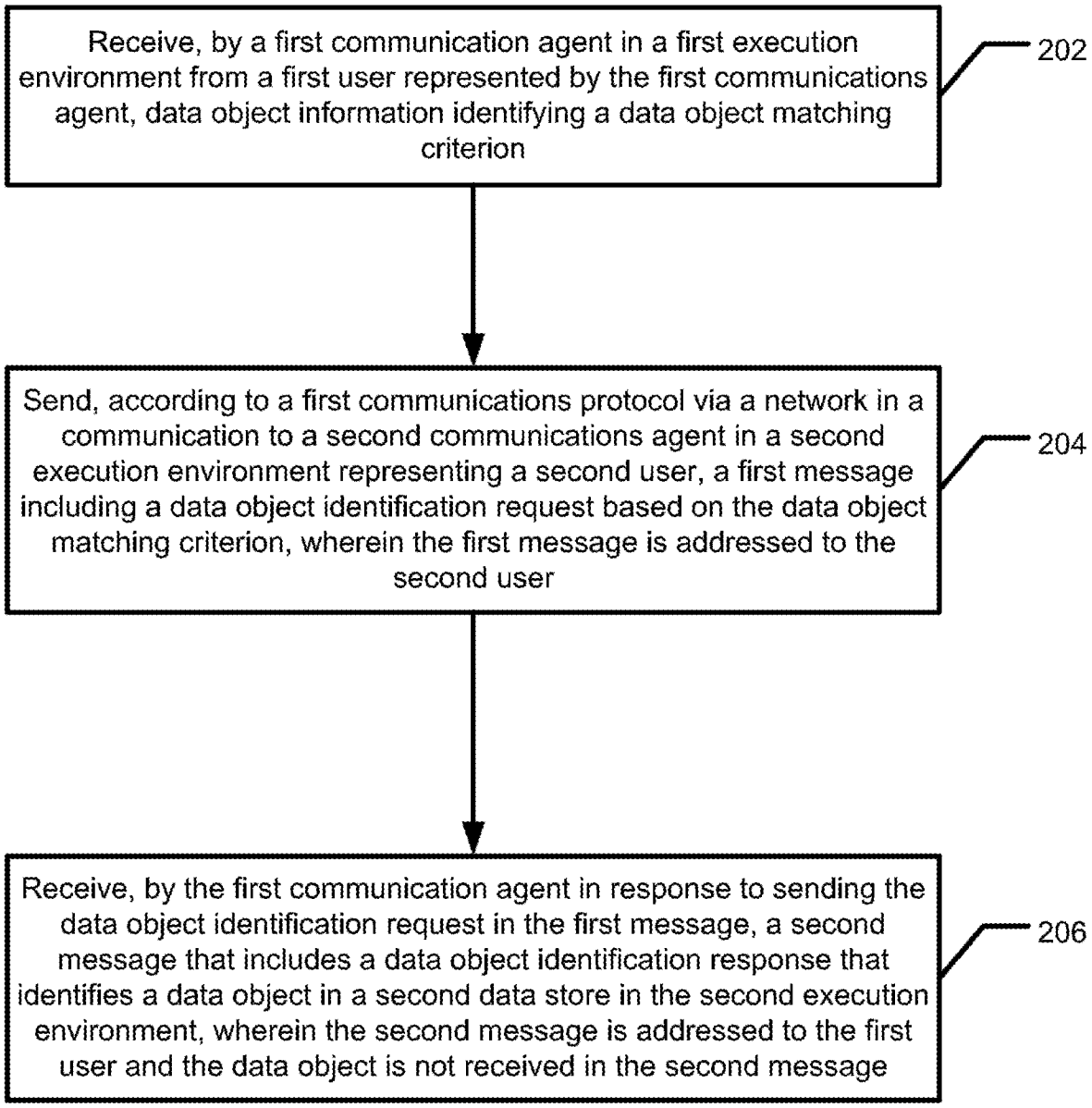

Receive, by a first communication agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion
202
Send, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user
204
Receive, by the first communication agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message
206

FIG. 2A

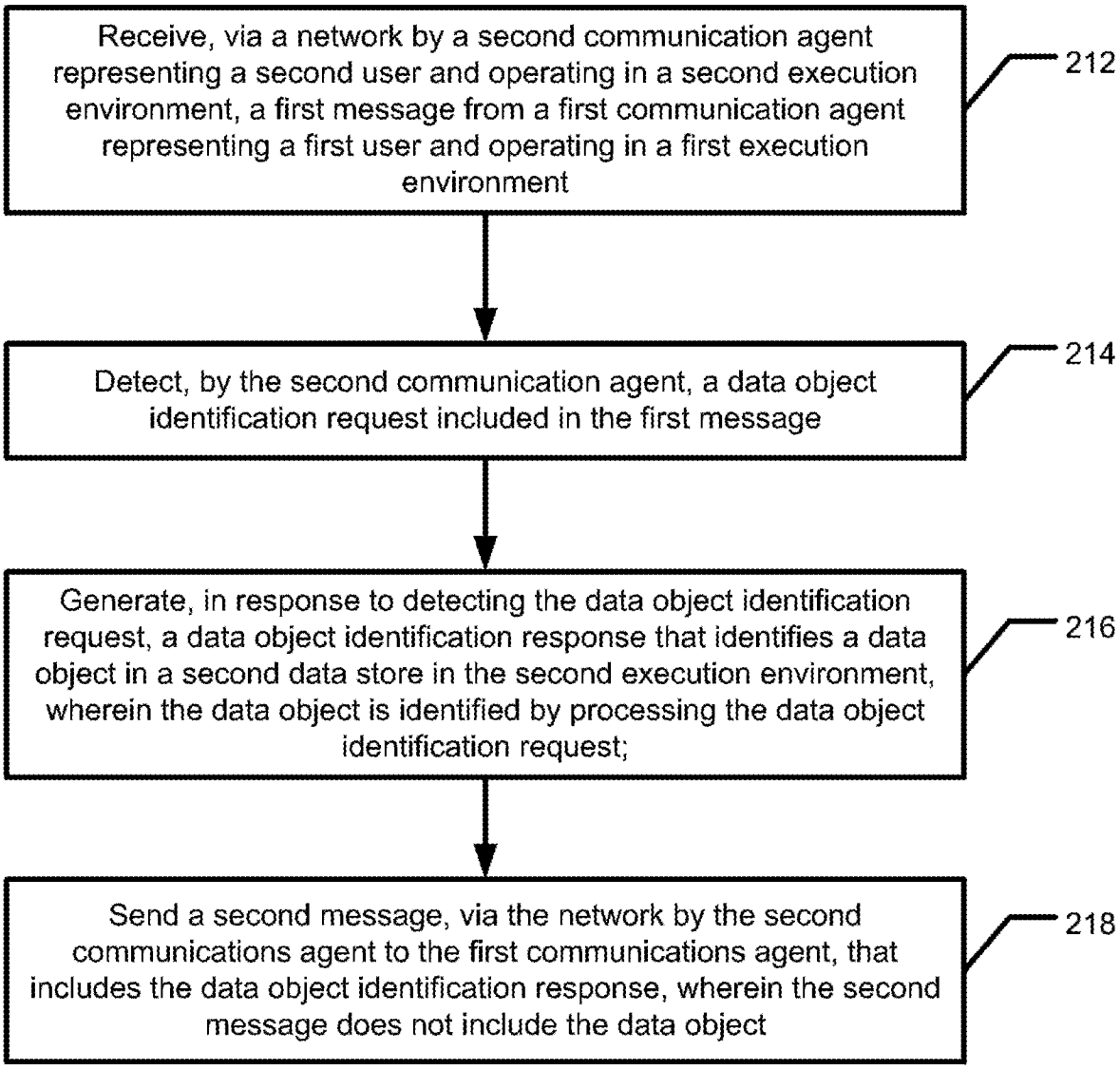

Receive, via a network by a second communication agent representing a second user and operating in a second execution environment, a first message from a first communication agent representing a first user and operating in a first execution environment
212
Detect, by the second communication agent, a data object identification request included in the first message
214
Generate, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request;
216
Send a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object
218

FIG. 2B

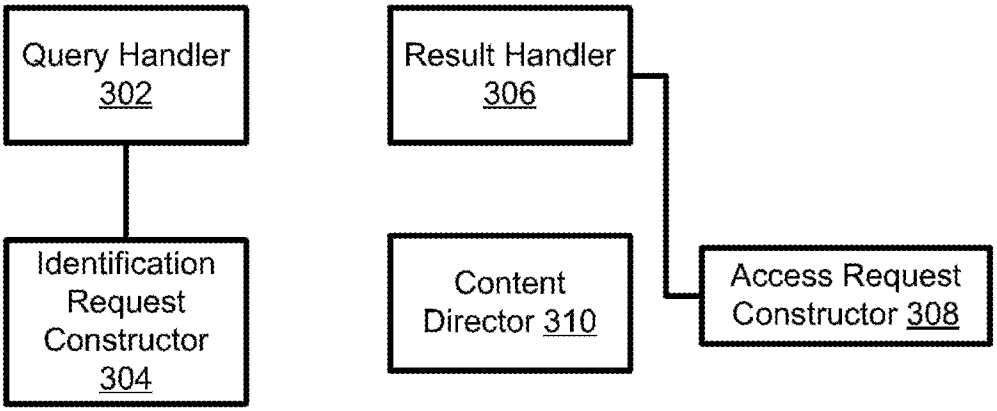

Query Handler 302
Result Handler 306
Identification Request Constructor 304
Content Director 310
Access Request Constructor 308

FIG. 3A

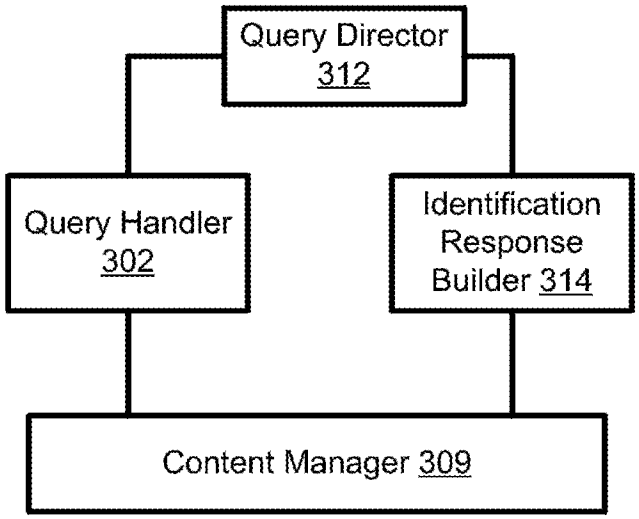

Query Director 312
Query Handler 302
Identification Response Builder 314
Content Manager 309

FIG. 3B

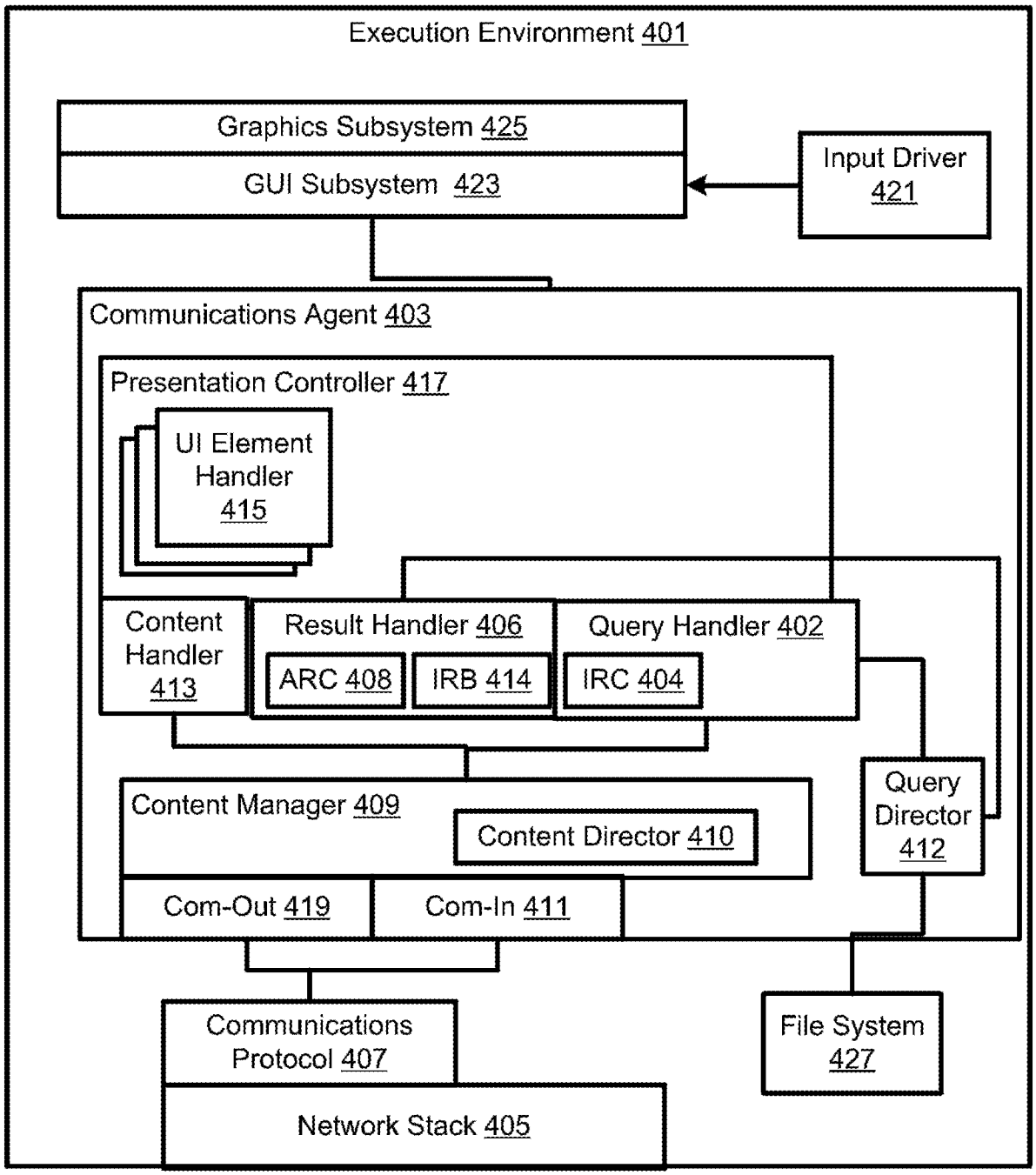

Execution Environment 401
Graphics Subsystem 425
GUI Subsystem 423
Input Driver 421
Communications Agent 403
Presentation Controller 417
UI Element Handler 415
Content Handler 413
Result Handler 406
ARC 408
IRB 414
Query Handler 402
IRC 404
Content Manager 409
Content Director 410
Query Director 412
Com-Out 419
Com-In 411
Communications Protocol 407
Network Stack 405
File System 427

FIG. 4

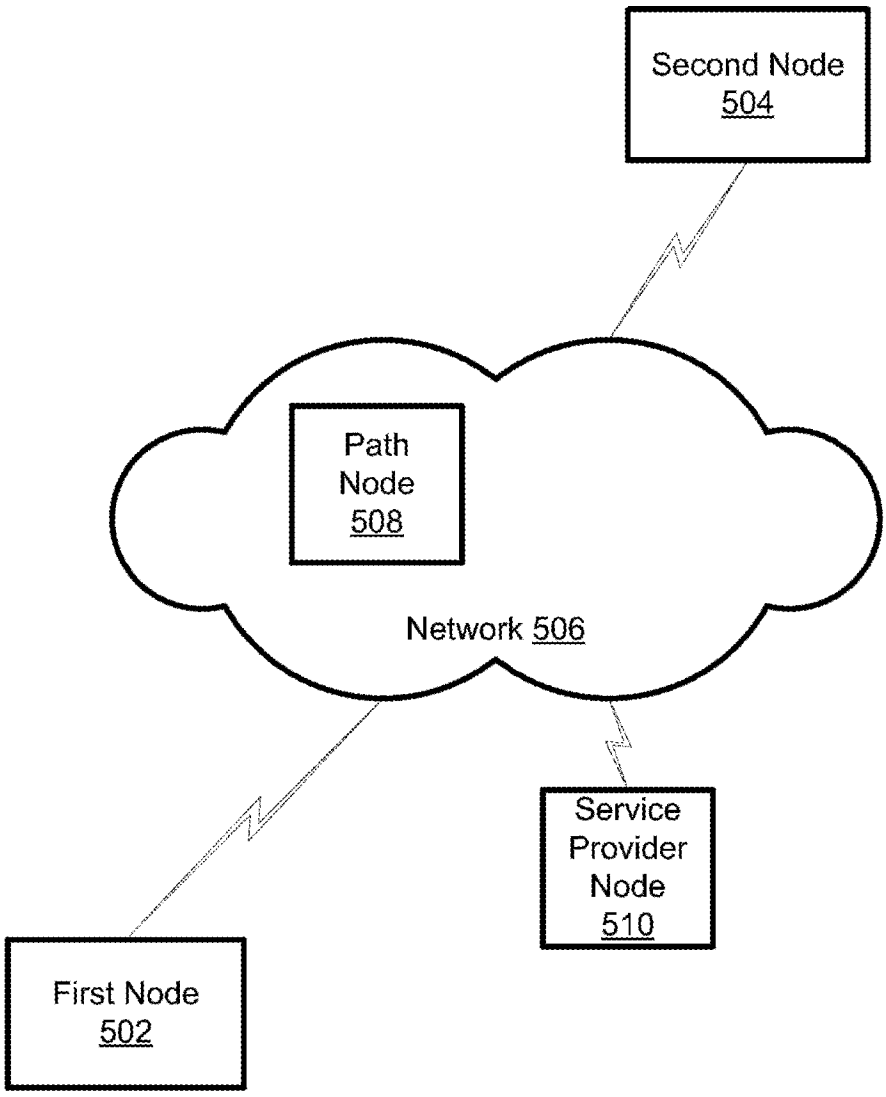

Second Node
504
Path
Node
508
Network 506
Service
Provider
Node
510
First Node
502

FIG. 5

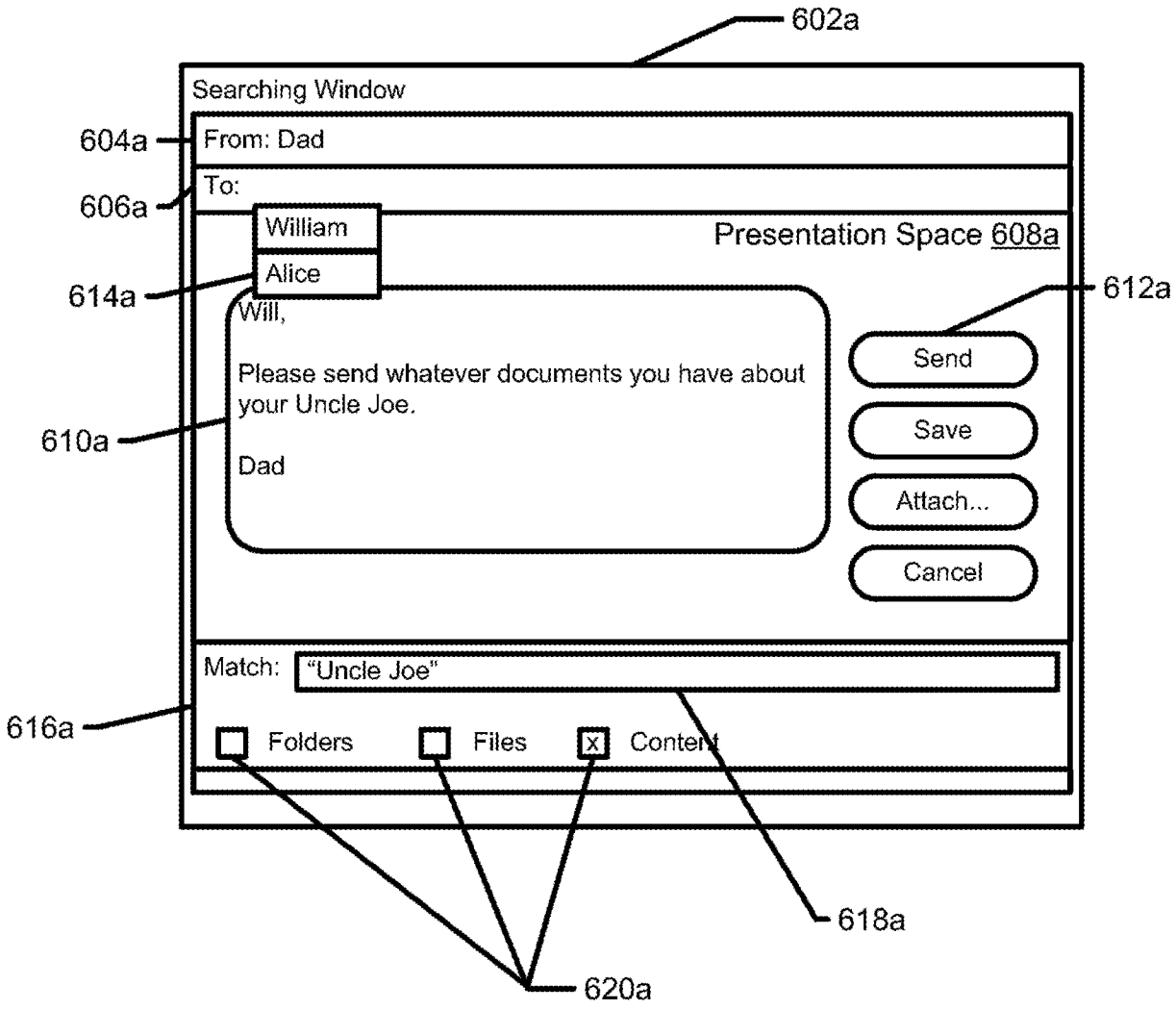

602a
Searching Window
604a
From: Dad
606a
To:
William
Alice
614a
Presentation Space 608a
612a
Will,
Please send whatever documents you have about your Uncle Joe.
610a
Dad
Send
Save
Attach...
Cancel
Match:
"Uncle Joe"
616a
Folders
Files
Content
618a
620a

FIG. 6A

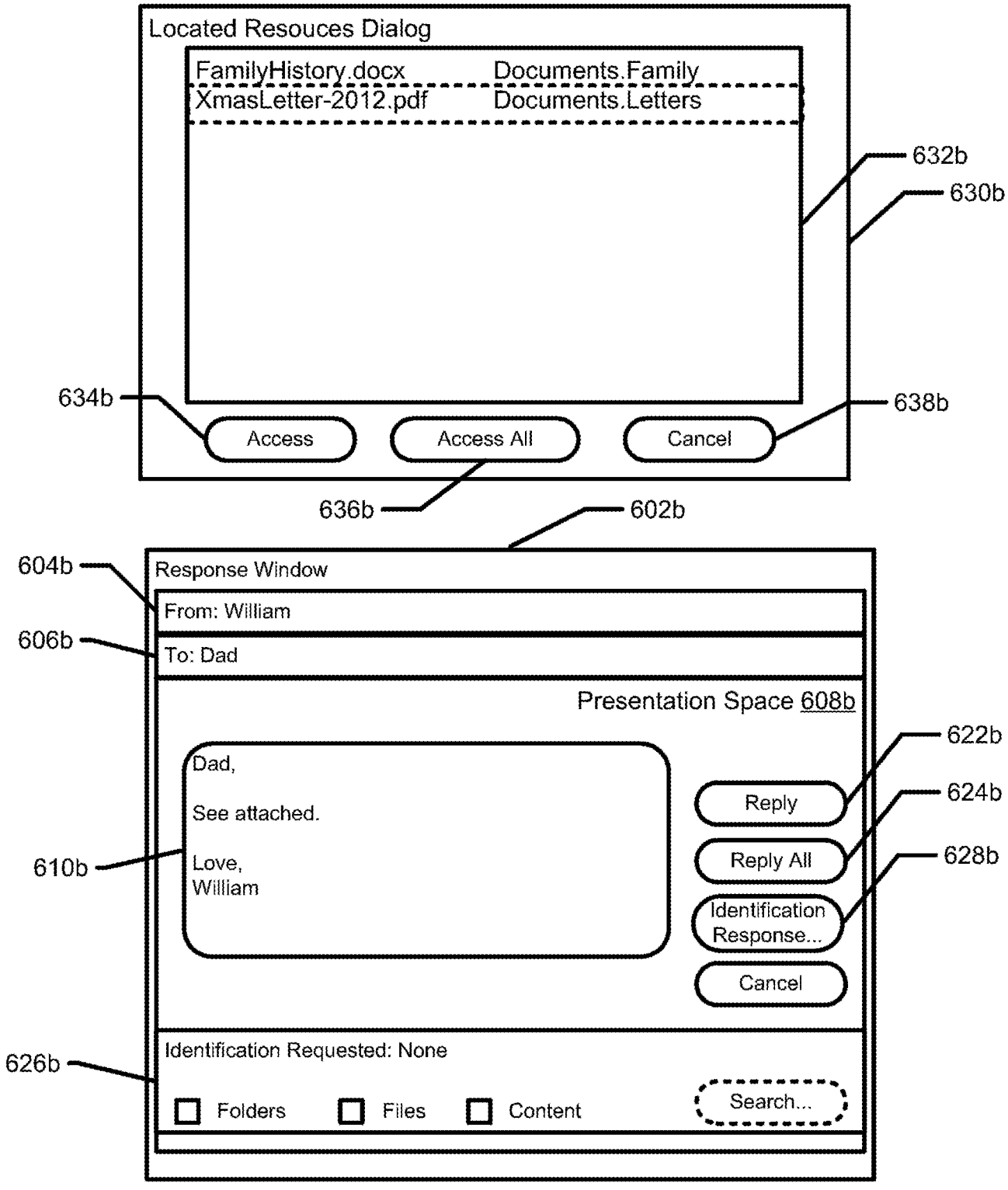

Located Resouces Dialog
FamilyHistory.docx Documents.Family
XmasLetter-2012.pdf Documents.Letters
632b
630b
634b
638b
Access
Access All
Cancel
636b
602b
604b
Response Window
From: William
606b
To: Dad
Presentation Space 608b
622b
624b
628b
610b
Dad,
See attached.
Love,
William
Reply
Reply All
Identification Response...
Cancel
626b
Identification Requested: None
Folders
Files
Content
Search...

FIG. 6B

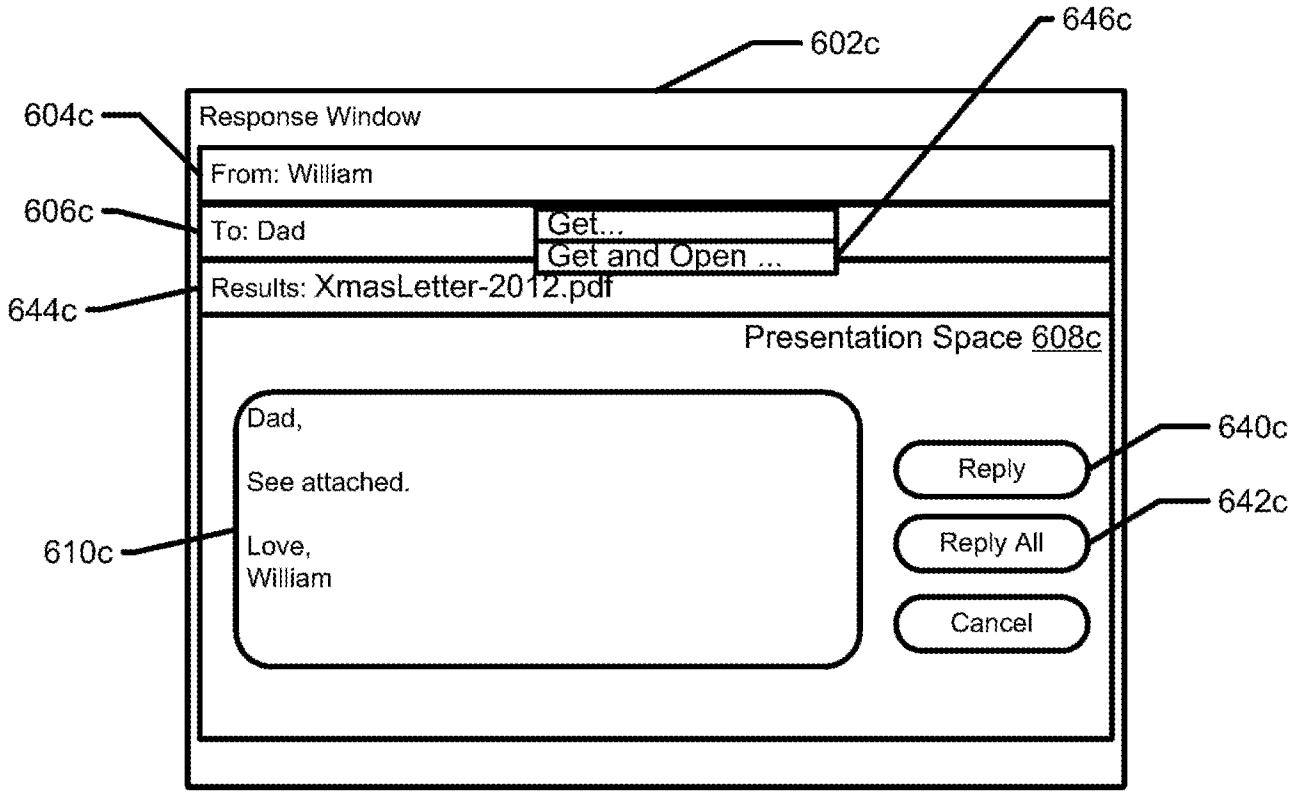

602c
646c
604c
Response Window
From: William
606c
To: Dad
Get...
Get and Open ...
644c
Results: XmasLetter-2012.pdf
Presentation Space 608c
Dad,
See attached.
Love,
William
610c
Reply
640c
Reply All
642c
Cancel

FIG. 6C

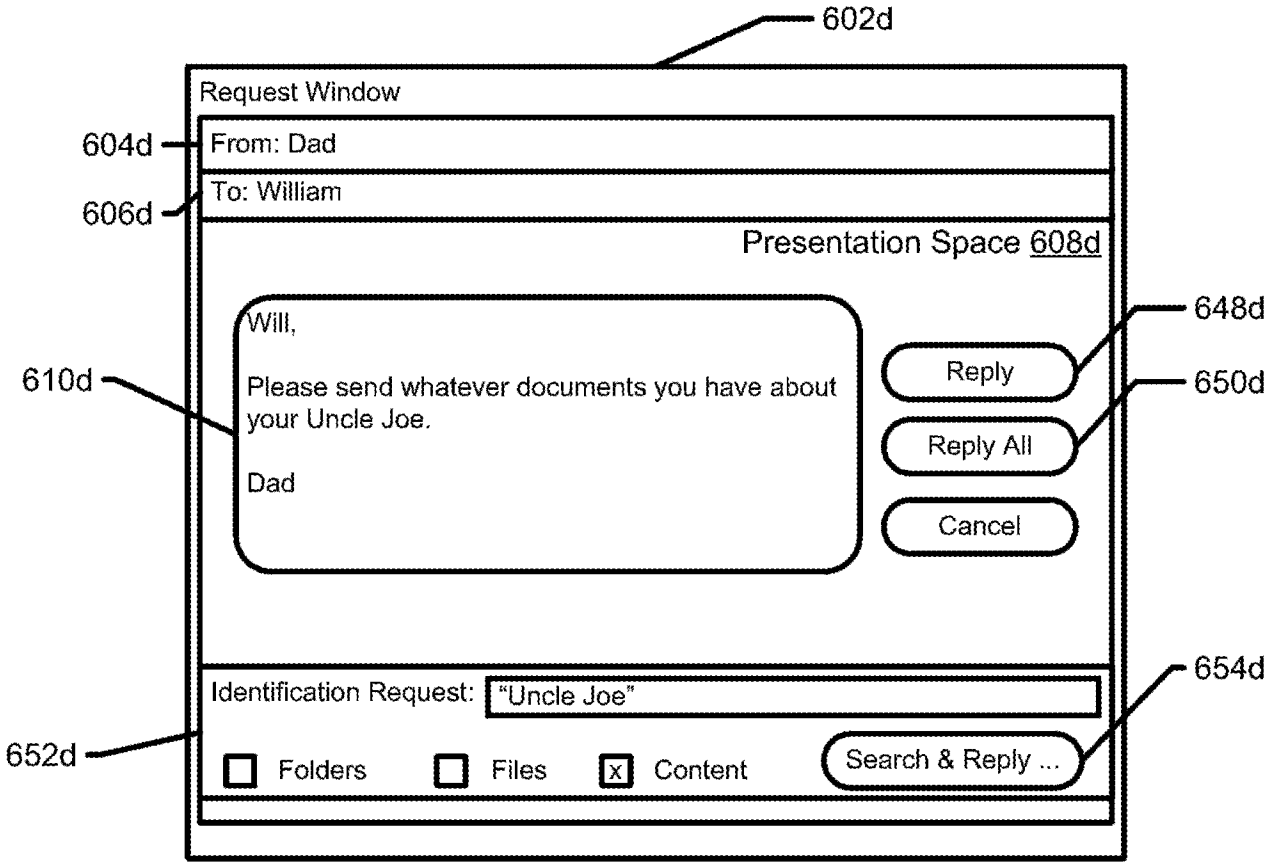

602d
Request Window
604d
From: Dad
606d
To: William
Presentation Space 608d
610d
Will,
Please send whatever documents you have about your Uncle Joe.
Dad
648d
Reply
650d
Reply All
Cancel
654d
Identification Request:
"Uncle Joe"
652d
Folders
Files
x
Content
Search & Reply ...

FIG. 6D

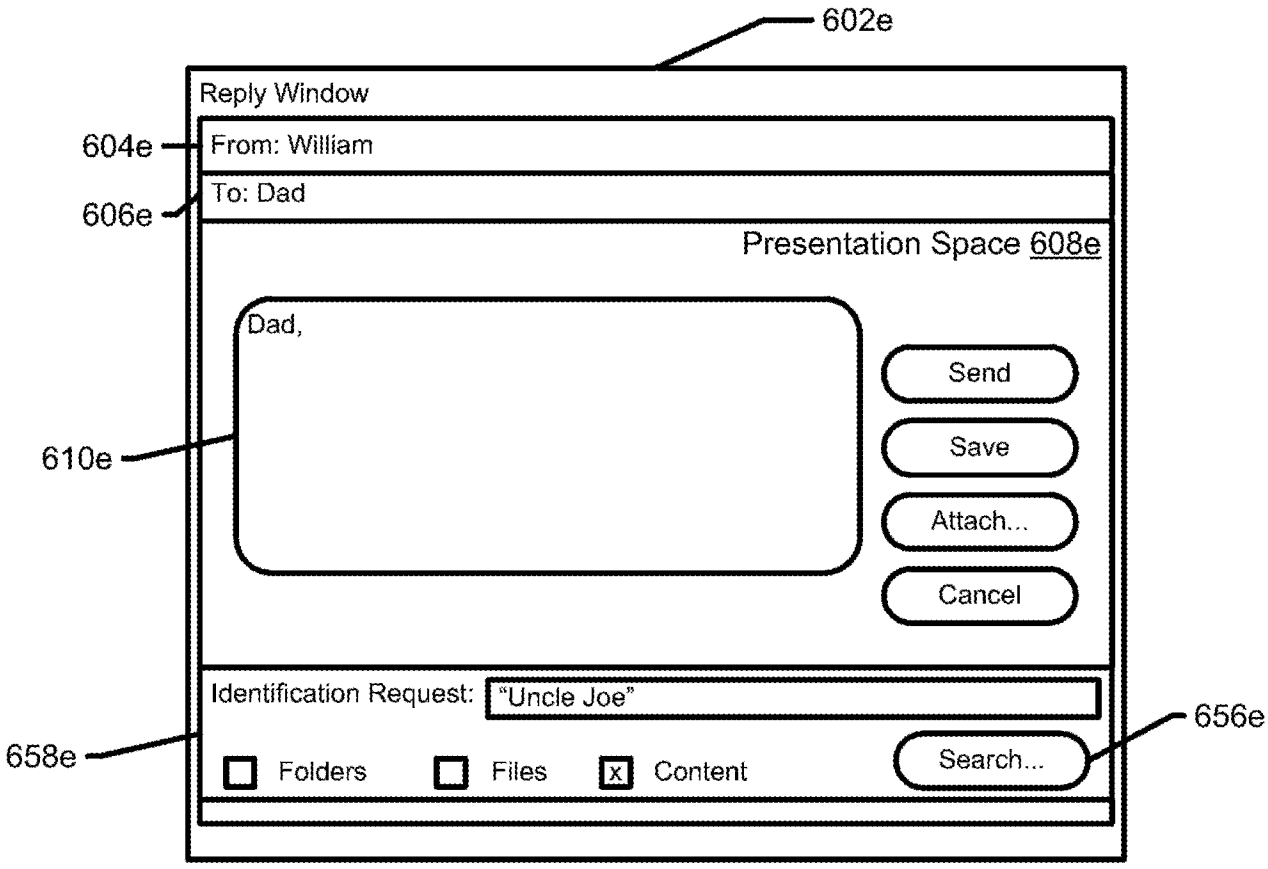

602e
Reply Window
604e
From: William
606e
To: Dad
Presentation Space 608e
Dad,
610e
Send
Save
Attach...
Cancel
Identification Request:
"Uncle Joe"
658e
656e
Folders
Files
x
Content
Search...

FIG. 6E

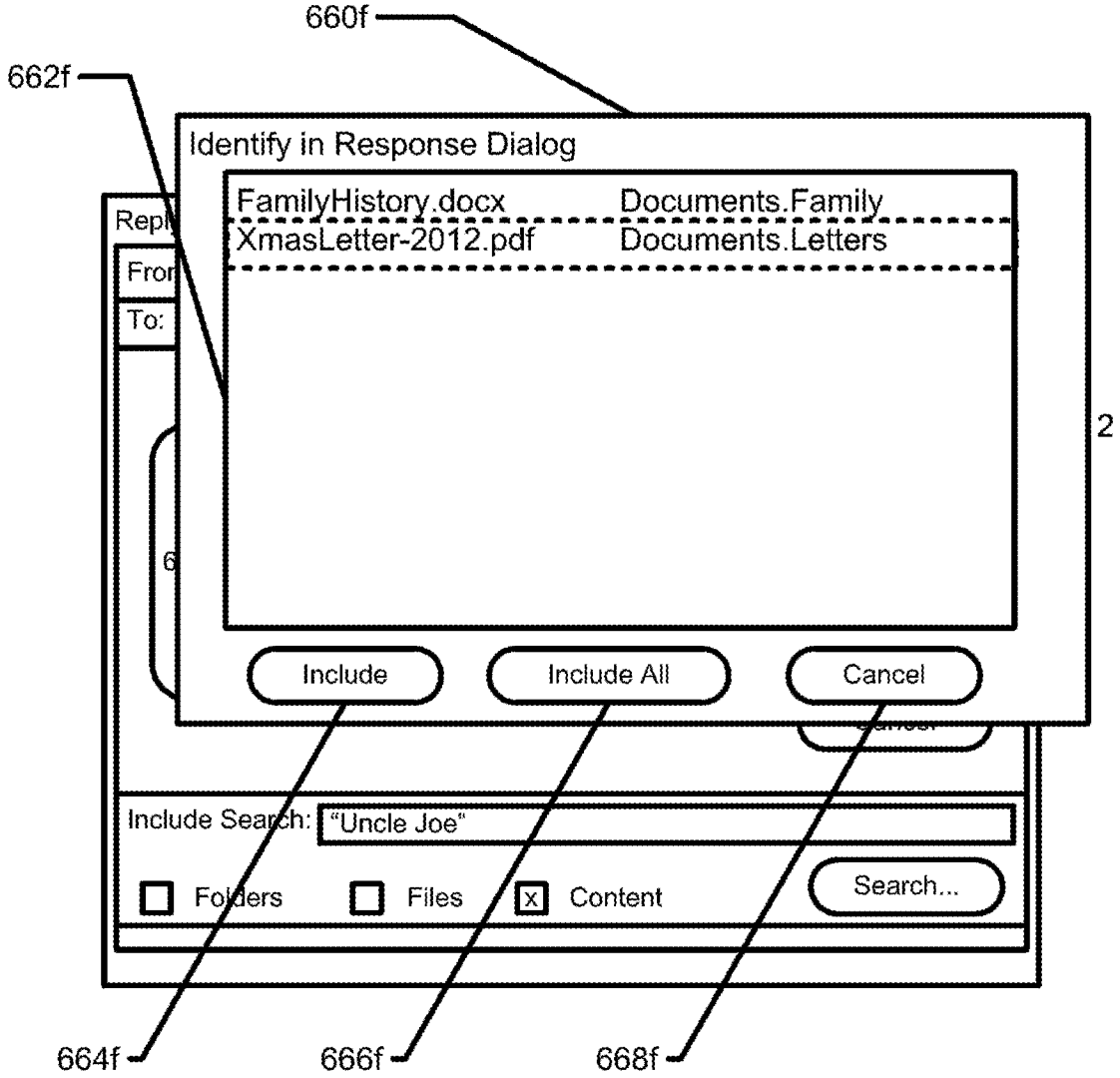

660f
662f
Identify in Response Dialog
FamilyHistory.docx Documents.Family
XmasLetter-2012.pdf Documents.Letters
Include
Include All
Cancel
Include Search: "Uncle Joe"
Folders
Files
Content
Search...
664f
666f
668f

FIG.6F

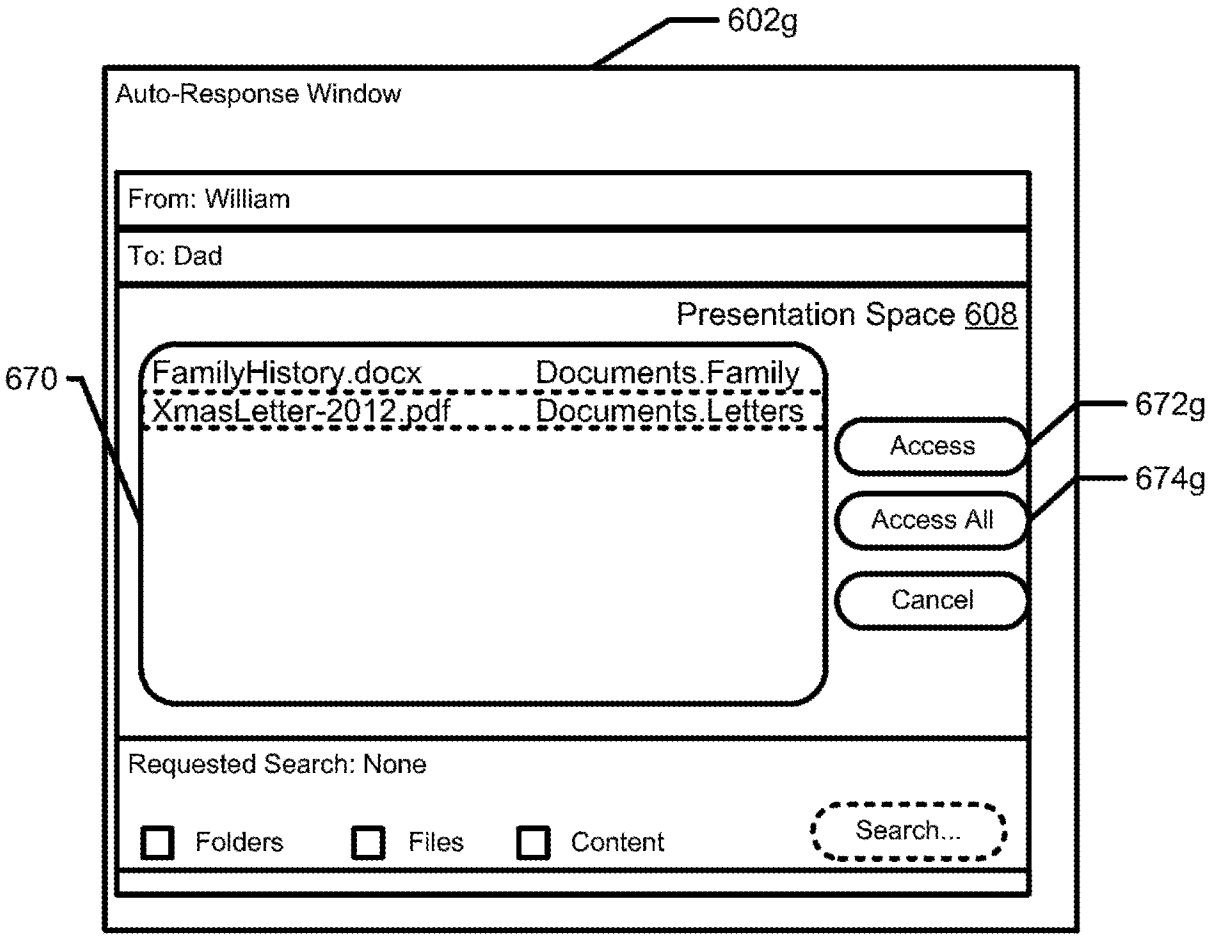

602g
Auto-Response Window
From: William
To: Dad
Presentation Space 608
670
FamilyHistory.docx Documents.Family
XmasLetter-2012.pdf Documents.Letters
Access
Access All
Cancel
672g
674g
Requested Search: None
Folders
Files
Content
Search...

FIG. 6G

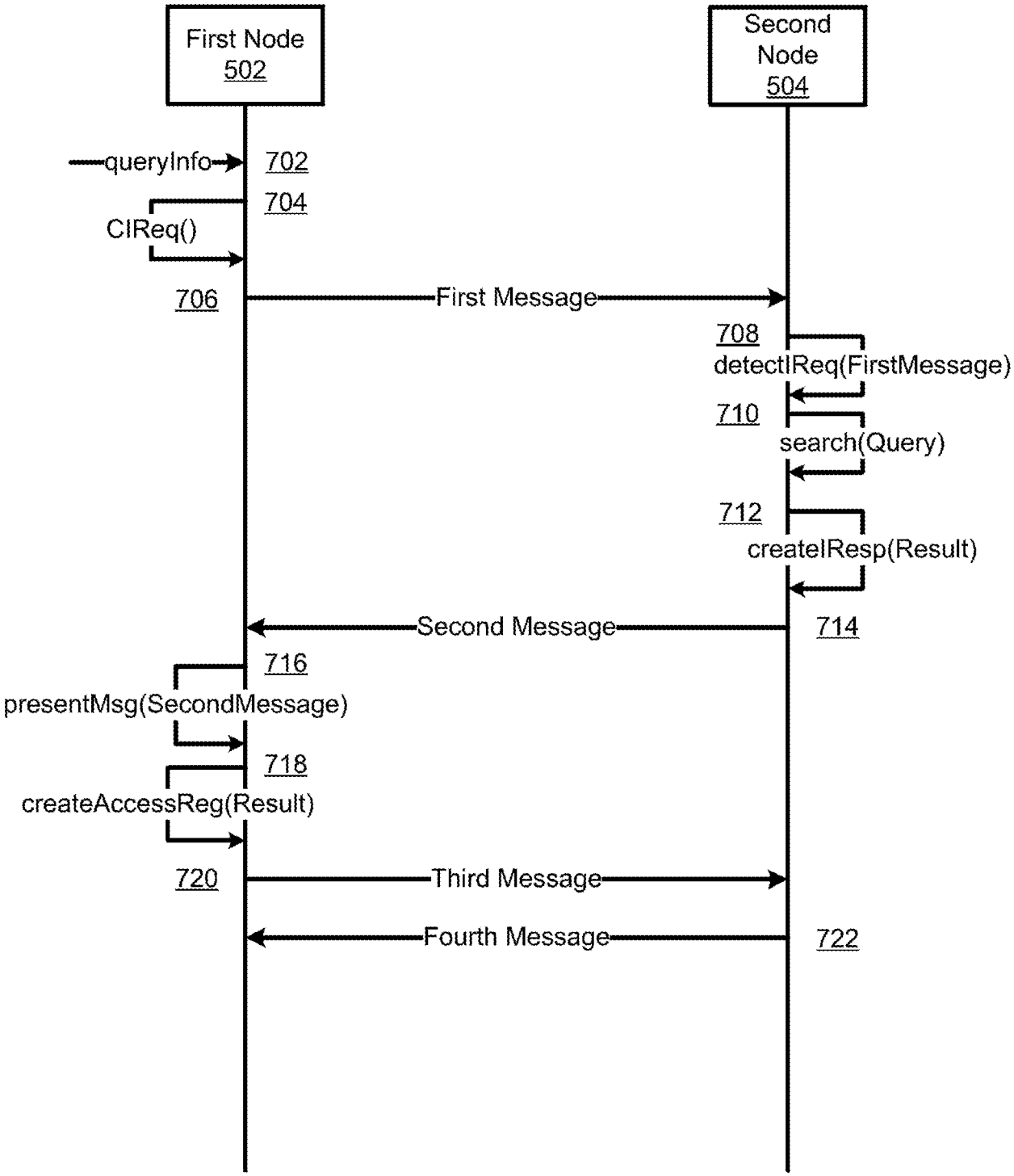

First Node
502
Second
Node
504
queryInfo
702
704
CIReq()
706
First Message
708
detectIReq(FirstMessage)
710
search(Query)
712
createIResp(Result)
Second Message
714
716
presentMsg(SecondMessage)
718
createAccessReg(Result)
720
Third Message
Fourth Message
722

FIG. 7

800a

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="--OoOT----toOo--------"

--OoOT----toOo--------
Content-Type: text/plain
Hi, this is the first message
--OoOT----toOo--------
Content-Type: application/id-request
Accept:  image/*
<id-request>
  <and>
    <criterion attribute="query" type="keyword" match-expression="'Uncle Joe'" />
    <criterion attribute="scope" resource-type="content" />
  </and>
</id-request>
--OoOT----toOo--------
```

814a, 812a, 804a, 806a, 810a, 808a, 802a

Fig. 8A

800b

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="--toOo----OoOt--------"

--toOo----OoOt---------
Content-Type: text/plain
Dad,

See attached.

Love,
William
--toOo----OoOt---------
Content-Type: application/query-respponse

FILE://Documents/Family/FamilyHistory.docx
FILE://Documents/Letters/XmasLetter-2012.pdf
--toOo----OoOt---------
```

802b, 806b, 804b

FIG. 8B

METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION

RELATED APPLICATIONS

[0001] The present application is a continuation-in-part of and claims priority to U.S. Patent Application No. 14/274,623 (published US 2014-0365588 A1) filed 05/09/2014 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION," which, in turn, is a continuation-in-part of and claims priority to U.S. Patent Application No. 13/624,906 (published US 2014-0089419 A1) filed 09/22/2012 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION" and U.S. Patent Application No. 13/626,635 (published under US 2014-0089421 A1) filed 09/25/2012 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR SHARING A DATA OBJECT IN A DATA STORE VIA A COMMUNICATION."

[0002] This application is related to the following U.S. Patent Applications:

[0003] U.S. Patent Application No. 12/833,016 (published US 2012-0011444 A1) filed on 2010/07/09, entitled “METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR REFERENCING AN ATTACHMENT IN A COMMUNICATION”;

[0004] U.S. Patent Application No. 12/833,014 (published US 2012-0011207 A1) filed on 2010/07/09, entitled “METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A REQUEST FOR A RESOURCE IN A COMMUNICATION”;

[0005] U.S. Patent Application No. 13/624,906 (published US 2014-0089419 A1) filed 2012/09/22 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION;” AND

[0006] U.S. Patent Application No. 13/626,635 (published under US 2014-0089421 A1) filed 2012/09/25 entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR

SHARING A DATA OBJECT IN A DATA STORE VIA A COMMUNICATION;" the entire contents of each are herein incorporated by reference for all purposes.

BACKGROUND

[0007] While receiving attachments in a network communication, such as email, is common, most attachments received are unsolicited or unrequested. To request an attachment, a user typically sends the request as voice and/or text data in a voice and/or text message heard and/or read by another user. The request may be as vague or as specific as the language used by the requesting user. The other user must interpret the request and find a file or other data object that seems to match the request. The other user in many cases locates a “matching” resource using a program or application other than the communications agent that received the request. For example, a search program may be used to search a hard-drive by the other user. The other user must enter the search criteria.

[0008] Accordingly, there exists a need for methods, systems, and computer program products for processing a data object identification request in a communication. .

SUMMARY

[0009] The following presents a simplified summary of the disclosure in order to provide a basic understanding to the reader. This summary is not an extensive overview of the disclosure and it does not identify key/critical elements of the invention or delineate the scope of the invention. Its sole purpose is to present some concepts disclosed herein in a simplified form as a prelude to the more detailed description that is presented later.

[0010] In one embodiment, an apparatus is provided, comprising: at least one non-transitory memory storing instructions; and one or more processors in communication with the at least one non-transitory memory, wherein the one or more processors execute the instructions to: cause, at a first node, display of at least one first interface with a first user interface element and a second user interface element, utilizing first hypertext markup language-equipped code that is sent via at least one network; receive, from the first node via the at least one network, a file description via the first user interface element that includes a textbox, utilizing the at least one first interface; receive, from the first node via the at least one network, an indication of at least one of a date or a time via the second user interface element, utilizing the at least one first interface; cause, at the first node, display of at least one second interface with a third user interface element and a fourth user interface element, utilizing second hypertext markup language-equipped code that is sent via the at least one network; receive, from the first node via the at least one network, indicia associated with at least one email address via the third user interface element, utilizing the at least one second interface; receive, from the first node via the at least one network, an indication of a selection of the fourth user interface element, utilizing the at least one second interface; in response to the receipt of the indication of the selection of the fourth user interface element and based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; generate a first message indicating the at least one of the date or the time, and including a fifth user interface element and at least one sentence including the file description; send, to a second node via the at least one network, the first message; receive, from the second node via the at least one

network, an indication of a selection of the fifth user interface element of the first message; in response to the receipt of the indication of the selection of the fifth user interface element, cause, at the second node, display of at least one third interface with a sixth user interface element and a seventh user interface element, utilizing third hypertext markup language-equipped code that is sent via at least one network; receive, from the second node via the at least one network, an indication of a selection at least one file utilizing the sixth user interface element, utilizing the at least one third interface; receive, from the second node via the at least one network, an indication of a selection of the seventh user interface element, utilizing the at least one third interface; in response to the receipt of the indication of the selection of the seventh user interface element, generate a second message that includes an eighth user interface element, and that does not include a file attachment with the second message; send, to the first node via the at least one network, the second message; receive, from the first node via the at least one network, an indication of a selection of the eighth user interface element of the second message; in response to the receipt of the indication of a selection of the eighth user interface element of the second message, cause, at the first node, display of at least one fourth interface with a reference to the at least one file for allowing access, at the first node, to the at least one file.

[0011] In another embodiment, an apparatus is provided, comprising at least one non-transitory memory storing instructions; and one or more processors in communication with the at least one non-transitory memory, wherein the one or more processors execute the instructions to: send a first communication over at least one network that results in display, at a first node, of a first user interface element, a second user interface element, and a third user interface element; receive, from the first node via the at least one network, text associated with one or more desired files via the first user interface element that includes a textbox; receive, from the first node via the at least one network, an object associated with at least one email address via the second user interface element; receive, from the first node via the at least one network, a message associated with a selection of the third user interface element; in response to the receipt of the message associated with the selection of the third user interface element and based on the text associated with the one or more desired files and the object

associated with the at least one email address; generate a first message including a fourth user interface element and the text associated with the one or more desired files; send, to a second node via the at least one network, the first message; receive, from the second node via the at least one network, a message associated with a selection of the fourth user interface element of the first message; in response to the receipt of the message associated with the selection of the fourth user interface element, send a third communication over the at least one network that results in display, at the second node, of a fifth user interface element; receive, from the second node via the at least one network, a message associated with a selection at least one file; receive, from the second node via the at least one network, a message associated with a selection of the fifth user interface element; in response to the receipt of the message associated with the selection of the fifth user interface element, generate a second message that includes an sixth user interface element, and that does not include a file attachment with the second message; send, to the first node via the at least one network, the second message; receive, from the first node via the at least one network, a message associated with a selection of the sixth user interface element of the second message; in response to the receipt of the message associated with the selection of the sixth user interface element of the second message, send a fourth communication over the at least one network that results in display, at the first node, of a reference to the at least one file; and allow access, at the first node, to the at least one file via the reference.

[0012] In yet another embodiment, a computer-implemented method is provided, comprising: displaying, at a first node, at least one first interface with a first user interface element and a second user interface element, utilizing first hypertext markup language-equipped code that is received via at least one network; receiving, utilizing the at least one first interface at the first node, a file description via the first user interface element that includes a textbox; sending, from the first node via the at least one network, the file description; receiving, utilizing the at least one first interface at the first node, an indication of at least one of a date or a time via the second user interface element; sending, from the first node via the at least one network, the indication of the at least one of the date or the time; displaying, at the first node, at least one second interface with a third user interface element and a fourth user interface element, utilizing

second hypertext markup language-equipped code that is received via the at least one network; receiving, utilizing the at least one second interface at the first node, indicia associated with at least one email address via the third user interface element; sending, from the first node via the at least one network, the indicia associated with the at least one email address; detecting, at the first node, a selection of the fourth user interface element; sending, from the first node via the at least one network, an indication of the selection of the fourth user interface element; after sending the indication of the selection of the fourth user interface element and based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; receiving, at a second node via the at least one network, a first message indicating the at least one of the date or the time, and including a fifth user interface element and at least one sentence including the file description; detecting, at the second node, a selection of the fifth user interface element of the first message; sending, from the second node via the at least one network, an indication of the selection of the fifth user interface element of the first message; after sending the indication of the selection of the fifth user interface element, displaying, at the second node, at least one third interface with a sixth user interface element and a seventh user interface element, utilizing third hypertext markup language-equipped code that is received via at least one network; receiving, utilizing the at least one third interface at the second node, a selection of at least one file utilizing the sixth user interface element; sending, from the second node via the at least one network, an indication of the at least one file; detecting, utilizing the at least one third interface at the second node, a selection of the seventh user interface element; sending, from the second node via the at least one network, an indication of the selection of the seventh user interface element; after sending the indication of the selection of the seventh user interface element, receiving, at the first node, a second message that includes an eighth user interface element, and that does not include a file attachment with the second message; detecting, at the first node, a selection of the eighth user interface element of the second message; sending, from the first node via the at least one network, an indication of the selection of the eighth user interface element of the second message; after sending the indication of a selection of the eighth user interface element of the second

message, displaying, at the first node, at least one fourth interface with a reference to the at least one file; detecting, utilizing the at least one fourth interface at the first node, a selection of the reference to the at least one file; and providing access, at the first node, to the at least one file, utilizing the reference.

[0013] In still yet another embodiment, an apparatus is provided, comprising: a first node configured to: display, at the first node, a first user interface element, a second user interface element, and a third user interface element; receive, at the first node, text associated with one or more desired files via the first user interface element that includes a textbox; send, from the first node via at least one network, the text associated with the one or more desired files; receive, at the first node, an object associated with at least one email address via the second user interface element; send, from the first node via the at least one network, the object associated with the at least one email address; detect, at the first node, a selection of the third user interface element; and send, from the first node via the at least one network, a message associated with the selection of the third user interface element; said first node further configured to communicate with a second node that is configured to: after the message associated with the selection of the third user interface element is sent and based on the text associated with the one or more desired files and the object associated with the at least one email address; receive, at the second node via the at least one network, a first message including a fourth user interface element and at least one sentence including the text associated with the one or more desired files; detect, at the second node, a selection of the fourth user interface element of the first message; send, from the second node via the at least one network, a message associated with the selection of the fourth user interface element of the first message; after the message associated with the selection of the fourth user interface element is sent, display, at the second node, a fifth user interface element and a sixth user interface element; detect, at the second node, a selection of at least one file; send, from the second node via the at least one network, a message associated with the at least one file; detect, at the second node, a selection of the sixth user interface element; and send, from the second node via the at least one network, a message associated with the selection of the sixth user interface element; said first node further configured to: after the message associated

with the selection of the sixth user interface element is sent, receive, at the first node, a second message that includes a seventh user interface element, and that does not include a file attachment with the second message; detect, at the first node, a selection of the seventh user interface element of the second message; send, from the first node via the at least one network, a message associated with the selection of the seventh user interface element of the second message; after the message associated with the selection of the seventh user interface element of the second message is sent, display, at the first node, an eighth user interface element associated with a reference to the at least one file; and provide access, at the first node, to the at least one file utilizing the reference.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] Objects and advantages of the present invention will become apparent to those skilled in the art upon reading this description in conjunction with the accompanying drawings, in which like reference numerals have been used to designate like or analogous elements, and in which:

[0015] FIG. **1** is a block diagram illustrating an exemplary hardware device included in and/or otherwise providing an execution environment in which the subject matter may be implemented;

[0016] FIG. **2A** is a flow diagram illustrating a method for processing a data object identification request in a communication according to an aspect of the subject matter described herein;

[0017] FIG. **2B** is a flow diagram illustrating a method for processing a data object identification request in a communication according to an aspect of the subject matter described herein;

[0018] FIG. **3A** is a block diagram illustrating an arrangement of components for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0019] FIG. **3B** is a block diagram illustrating an arrangement of components for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0020] FIG. **4** is a block diagram illustrating an arrangement of components for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0021] FIG. **5** is a network diagram illustrating a system for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0022] FIG. **6A** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0023] FIG. **6B** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0024] FIG. **6C** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0025] FIG. **6D** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0026] FIG. **6E** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0027] FIG. **6F** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0028] FIG. **6G** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0029] FIG. **7** is a message flow diagram illustrating an exemplary data and execution flow for processing a data object identification request in a communication according to an aspect of the subject matter described herein;

[0030] FIG. **8A** illustrates an exemplary portion of a communication between a first execution environment and a second execution environment according to an aspect of the subject matter described herein; and

[0031] FIG. **8B** illustrates another exemplary portion of a communication between a first execution environment and a second execution environment according to an aspect of the subject matter described herein.

DETAILED DESCRIPTION

[0032] One or more aspects of the disclosure are described with reference to the drawings, wherein like reference numerals are generally utilized to refer to like elements throughout, and wherein the various structures are not necessarily drawn to scale. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of one or more aspects of the disclosure. It may be evident, however, to one skilled in the art, that one or more aspects of the disclosure may be practiced with a lesser degree of these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate describing one or more aspects of the disclosure.

[0033] Unless otherwise defined, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this disclosure belongs. Although methods, components, and devices similar or equivalent to those described herein can be used in the practice or testing of the subject matter described herein, suitable methods, components, and devices are described below.

[0034] All publications, patent applications, patents, and other references mentioned herein are incorporated by reference in their entirety. In case of conflict, the present specification, including definitions, will control. In addition, the materials, methods, and examples are illustrative only and not intended to be limiting.

[0035] An exemplary device included in an execution environment that may be configured according to the subject matter is illustrated in FIG. **1**. An "execution environment", as used herein, is an arrangement of hardware and, in some aspects, software that may be further configured to include and/or otherwise host an arrangement of components for performing a method of the subject matter described herein. An execution environment includes and/or is otherwise provided by one or more devices. The execution environment is said to be the execution environment of the device and/or devices. An execution environment may be and/or may include a virtual execution environment including software components operating in a host execution environment. Exemplary devices included in and/or otherwise providing suitable execution environments for configuring according to the subject matter include personal computers, notebook computers, tablet computers, servers, portable electronic devices, handheld electronic devices, mobile devices, multiprocessor devices, distributed systems, consumer electronic devices, routers, communication servers, and/or any other suitable devices. Those skilled in the art will understand that the components illustrated in FIG. **1** are exemplary and may vary by particular execution environment.

[0036] FIG. **1** illustrates a hardware device **100** included in an execution environment **102**. FIG. **1** illustrates that execution environment **102** includes a processor **104**, such as one or more microprocessors; a physical processor memory **106** including storage locations identified by addresses in a physical memory address space of processor **104**; a persistent secondary storage **108**, such as one or more hard drives and/or flash

storage media; an input device adapter **110**, such as a key or keypad hardware, a keyboard adapter, and/or a mouse adapter; an output device adapter **112**, such as a display and/or an audio adapter for presenting information to a user; a network interface component, illustrated by a network interface adapter **114**, for communicating via a network such as a LAN and/or WAN; and a communication mechanism that operatively couples elements **104-114**, illustrated as a bus **116**. Elements **104-114** may be operatively coupled by various means. Bus **116** may comprise any type of bus architecture, including a memory bus, a peripheral bus, a local bus, and/or a switching fabric.

[0037] As used herein a "processor" is an instruction execution machine, apparatus, or device. A processor may include one or more electrical, optical, and/or mechanical components that operate in interpreting and executing program instructions. Exemplary processors include one or more microprocessors, digital signal processors (DSPs), graphics processing units, application-specific integrated circuits (ASICs), optical or photonic processors, and/or field programmable gate arrays (FPGAs). Processor **104** may access machine code instructions and data via one or more memory address spaces in addition to the physical memory address space. A memory address space includes addresses identifying locations in a processor memory. The addresses in a memory address space are included in defining a processor memory. Processor **104** may have more than one processor memory. Thus, processor **104** may have more than one memory address space. Processor **104** may access a location in a processor memory by processing an address identifying the location. The processed address may be identified by an operand of a machine code instruction and/or may be identified by a register or other portion of processor **104**.

[0038] FIG. **1** illustrates a virtual processor memory **118** spanning at least part of physical processor memory **106** and may span at least part of persistent secondary storage **108**. Virtual memory addresses in a memory address space may be mapped to physical memory addresses identifying locations in physical processor memory **106**. An address space for identifying locations in a virtual processor memory is referred to as a virtual memory address space; its addresses are referred to as virtual memory addresses; and its processor memory is referred to as a virtual processor memory or

virtual memory. The term "processor memory" may refer to physical processor memory, such as processor memory **106**, and/or may refer to virtual processor memory, such as virtual processor memory **118**, depending on the context in which the term is used.

[0039] Physical processor memory **106** may include various types of memory technologies. Exemplary memory technologies include static random access memory (SRAM) and/or dynamic RAM (DRAM) including variants such as dual data rate synchronous DRAM (DDR SDRAM), error correcting code synchronous DRAM (ECC SDRAM), RAMBUS DRAM (RDRAM), and/or XDR™ DRAM. Physical processor memory **106** may include volatile memory as illustrated in the previous sentence and/or may include nonvolatile memory such as nonvolatile flash RAM (NVRAM) and/or ROM.

[0040] Persistent secondary storage **108** may include one or more flash memory storage devices, one or more hard disk drives, one or more magnetic disk drives, and/or one or more optical disk drives. Persistent secondary storage may include a removable data storage medium. The drives and their associated tangible computer readable storage media provide volatile and/or nonvolatile storage for computer-readable instructions, data structures, program components, and other data for execution environment **102**.

[0041] Execution environment **102** may include software components stored in persistent secondary storage **108**, in remote storage accessible via a network, and/or in a processor memory. FIG. **1** illustrates execution environment **102** including an operating system **120**, one or more applications **122**, and other program code and/or data components illustrated by other libraries and subsystems **124**. In an aspect, some or all software components may be stored in locations accessible to processor **104** in a shared memory address space shared by the software components. The software components accessed via the shared memory address space are stored in a shared processor memory defined by the shared memory address space. In another aspect, a first software component may be stored in one or more locations accessed by processor **104** in a first address space and a second software component may be stored in one or more locations accessed by processor **104** in a second address space. The first software component is stored in a first processor memory defined by the first address

space and the second software component is stored in a second processor memory defined by the second address space.

[0042] Software components typically include instructions executed by processor **104** in a computing context referred to as a "process". A process may include one or more "threads". A "thread" includes a sequence of instructions executed by processor **104** in a computing sub-context of a process. The terms "thread" and "process" may be used interchangeably herein when a process includes only one thread.

[0043] Execution environment **102** may receive user-provided information via one or more input devices illustrated by an input device **128**. Input device **128** provides input information to other components in execution environment **102** via input device adapter **110**. Execution environment **102** may include an input device adapter for a keyboard, a touch screen, a microphone, a joystick, a television receiver, a video camera, a still camera, a document scanner, a fax, a phone, a modem, a network interface adapter, and/or a pointing device, to name a few exemplary input devices.

[0044] Input device **128** included in execution environment **102** may be included in device **100** as FIG. **1** illustrates or may be external (not shown) to device **100**. Execution environment **102** may include one or more internal and/or external input devices. External input devices may be connected to device **100** via corresponding communication interfaces such as a serial port, a parallel port, and/or a universal serial bus (USB) port. Input device adapter **110** receives input and provides a representation to bus **116** to be received by processor **104**, physical processor memory **106**, and/or other components included in execution environment **102**.

[0045] An output device **130** in FIG. **1** exemplifies one or more output devices that may be included in and/or that may be external to and operatively coupled to device **100**. For example, output device **130** is illustrated connected to bus **116** via output device adapter **112**. Output device **130** may be a display device. Exemplary display devices include liquid crystal displays (LCDs), light emitting diode (LED) displays, and projectors. Output device **130** presents output of execution environment **102** to one or more users. In some embodiments, an input device may also include an output device. Examples include a phone, a joystick, and/or a touch screen. In addition to various types of display devices, exemplary output devices include printers, speakers, tactile

output devices such as motion-producing devices, and other output devices producing sensory information detectable by a user. Sensory information detected by a user is referred herein to as "sensory input" with respect to the user.

[0046] A device included in and/or otherwise providing an execution environment may operate in a networked environment communicating with one or more devices via one or more network interface components. The terms "communication interface component" and "network interface component" are used interchangeably herein. FIG. **1** illustrates network interface adapter (NIA) **114** as a network interface component included in execution environment **102** to operatively couple device **100** to a network. A network interface component includes a network interface hardware (NIH) component and optionally a network interface software (NIS) component.

[0047] Exemplary network interface components include network interface controller components, network interface cards, network interface adapters, and line cards. A node may include one or more network interface components to interoperate with a wired network and/or a wireless network. Exemplary wireless networks include a BLUETOOTH network, a wireless 802.11 network, and/or a wireless telephony network (e.g., a cellular, PCS, CDMA, and/or GSM network). Exemplary network interface components for wired networks include Ethernet adapters, Token-ring adapters, FDDI adapters, asynchronous transfer mode (ATM) adapters, and modems of various types. Exemplary wired and/or wireless networks include various types of LANs, WANs, and/or personal area networks (PANs). Exemplary networks also include intranets and internets such as the Internet.

[0048] The terms "network node" and "node" in this document both refer to a device having a network interface component for operatively coupling the device to a network. Further, the terms "device" and "node" used herein may refer to one or more devices and nodes, respectively, providing and/or otherwise included in an execution environment unless clearly indicated otherwise.

[0049] The user-detectable outputs of a user interface are generically referred to herein as "user interface elements" or abbreviated as "UI elements". More specifically, visual outputs of a user interface are referred to herein as "visual interface elements". A visual interface element may be a visual output of a graphical user interface (GUI).

Exemplary visual interface elements include windows, textboxes, sliders, list boxes, drop-down lists, spinners, various types of menus, toolbars, ribbons, combo boxes, tree views, grid views, navigation tabs, scrollbars, labels, tooltips, text in various fonts, balloons, dialog boxes, and various types of button controls including check boxes and radio buttons. An application interface may include one or more of the elements listed. Those skilled in the art will understand that this list is not exhaustive. The terms “visual representation”, “visual output”, and “visual interface element” are used interchangeably in this document. Other types of UI elements include audio outputs referred to as ”audio interface elements”, tactile outputs referred to as “tactile interface elements”, and the like.

[0050] A visual output may be presented in a two-dimensional presentation where a location may be defined in a two-dimensional space having a vertical dimension and a horizontal dimension. A location in a horizontal dimension may be referenced according to an X-axis and a location in a vertical dimension may be referenced according to a Y-axis. In another aspect, a visual output may be presented in a three-dimensional presentation where a location may be defined in a three-dimensional space having a depth dimension in addition to a vertical dimension and a horizontal dimension. A location in a depth dimension may be identified according to a Z-axis. A visual output in a two-dimensional presentation may be presented as if a depth dimension existed allowing the visual output to overlie and/or underlie some or all of another visual output.

[0051] An order of visual outputs in a depth dimension is herein referred to as a "Z-order". The term “Z-value” as used herein refers to a location in a Z-order. A Z-order specifies the front-to-back and/or back-to-front ordering of visual outputs in a presentation space with respect to a Z-axis. In one aspect, a visual output with a higher Z-value than another visual output may be defined to be on top of or closer to the front than the other visual output. In another aspect, a visual output with a lower Z-value than another visual output may be defined to be on top of or closer to the front than the other visual output.

[0052] A “user interface (UI) element handler” component, as the term is used herein, includes a component of configured to send information representing a program entity for presenting a user-detectable representation of the program entity by an output

device, such as a display. A “program entity” is an object included in and/or otherwise processed by an application or executable. The user-detectable representation is presented based on the sent information. Information that represents a program entity for presenting a user detectable representation of the program entity by an output device is referred to herein as “presentation information”. Presentation information may include and/or may otherwise identify data in one or more formats. Exemplary formats include image formats such as raw pixel data, JPEG, video formats such as MP4, markup language data such as hypertext markup language (HTML) and other XML-based markup, a bit map, and/or instructions such as those defined by various script languages, byte code, and/or machine code. For example, a web page received by a user agent from a remote application provider may include HTML, ECMAScript, and/or byte code for presenting one or more UI elements included in a user interface of the remote application. Components configured to send information representing one or more program entities for presenting particular types of output by particular types of output devices include visual interface element handler components, audio interface element handler components, tactile interface element handler components, and the like.

[0053] A representation of a program entity may be stored and/or otherwise maintained in a presentation space. As used in this document, the term “presentation space” refers to a storage region allocated and/or otherwise provided for storing presentation information, which may include audio, visual, tactile, and/or other sensory data for presentation by and/or on an output device. For example, a buffer for storing an image and/or text string may be a presentation space as sensory information for a user. A presentation space may be physically and/or logically contiguous or non-contiguous. A presentation space may have a virtual as well as a physical representation. A presentation space may include a storage location in a processor memory, secondary storage, a memory of an output adapter device, and/or a storage medium of an output device. A screen of a display, for example, is a presentation space.

[0054] As used herein, the terms "program” and "executable” refer to any data representation that may be translated into a set of machine code instructions and may optionally include associated program data. The terms are used interchangeably herein.

Program representations other than machine code include object code, byte code, and source code. Object code includes a set of instructions and/or data elements that either are prepared for linking prior to loading or are loaded into an execution environment. When in an execution environment, object code may include references resolved by a linker and/or may include one or more unresolved references. The context in which this term is used will make clear the state of the object code when it is relevant. This definition can include machine code and virtual machine code, such as Java™ byte code. As used herein, the terms “application”, and “service” may be realized in one or more executables and/or in one or more hardware components. The terms are used interchangeably herein.

[0055] As used herein, an “addressable entity” is program or a portion thereof, specifiable in programming language in source code. An addressable entity is addressable in a program component translated for a compatible execution environment from the source code. Examples of addressable entities include variables, constants, functions, subroutines, procedures, modules, methods, classes, objects, code blocks, and labeled instructions. A code block includes one or more instructions in a given scope specified in a programming language. An addressable entity may include a value. In some places in this document “addressable entity” refers to a value of an addressable entity. In these cases, the context will clearly indicate that the value is being referenced.

[0056] Addressable entities may be written in and/or translated to a number of different programming languages and/or representation languages, respectively. An addressable entity may be specified in and/or translated into source code, object code, machine code, byte code, and/or any intermediate languages for processing by an interpreter, compiler, linker, loader, and/or other analogous tool.

[0057] As used herein, the term “network protocol” refers to a formal set of rules, conventions and data structures that governs how computers and other network devices exchange information over a network. The rules, conventions, and data structures are said to be specified or defined in a specification and/or schema.

[0058] The term "network path" as used herein refers to a sequence of nodes in a network that are communicatively coupled for transmitting data in one or more data units of a specified network protocol between a pair of nodes in the network.

[0059] A "data unit", as the term is used herein, is a data entity specified according to a network protocol for transmitting data contained in and/or otherwise identified by the data entity. A data unit is transmitted between a pair of nodes in a network path to send the data in and/or otherwise identified by the data unit from a source node to a destination node that includes a protocol endpoint of the network protocol. A network protocol explicitly and/or implicitly specifies and/or otherwise identifies a schema that defines one or more of a format for a valid data unit and a vocabulary for content of a valid data unit. One example of a data unit is an Internet Protocol (IP) packet. The Internet Protocol defines a format for an IP packet that defines a header for identifying a destination address that identifies a destination node and a payload portion for including a representation of data to be delivered to the identified destination node. The terms "data unit", "data packet", and "packet" are used interchangeably herein. One or more data units of a first network protocol may transmit a message of second network protocol. For example, one or more data units of the IP protocol may include a TCP message. In another example, one or more TCP data units may transmit an HTTP message.

[0060] How data is packaged in one more data units for a network protocol may vary as the data traverses a network path from a source node to a destination node. Data may be transmitted in a single data unit between two consecutive nodes in a network path. Additionally, data may be exchanged between a pair of consecutive nodes in several data units each including a portion of the data. Data received in a single data unit by a node in a network path may be split into portions included in several respective data units for transmitting to a next node in the network path. Portions of data received in several data units may be combined into a single data unit for transmitting by a node in a network path. For purposes of describing the subject matter, a data unit in which data is received by a node is referred to as a different data unit than a data unit in which the data is forwarded by the node.

[0061] A “protocol address”, as the term is used herein, for a network protocol is an identifier of a protocol endpoint that may be represented in a data unit of the protocol. For example 192.168.1.1 is an IP protocol address represented in a human readable format that may be included in an address portion of an IP header to identify a source and/or a destination IP protocol endpoint. A protocol address differs from a symbolic identifier, defined below, in that a symbolic identifier, with respect to a network protocol, maps to a protocol address. Thus, “www. mynode.com” may be a symbolic identifier for a node in a network when mapped to the protocol address 192.168.1.1. An identifier may be both a symbolic identifier and a protocol address depending on its role with respect to its use for a particular network protocol.

[0062] Since a protocol endpoint is included in a node and is accessible via a network via a network interface, a protocol address identifies a node and identifies a network interface of the node. A network interface may include one or more NICs operatively coupled to a network.

[0063] A node in a pair of nodes in a network path at one end of the sequence of nodes in the network path and/or the other end is referred to herein as a “path end node”. Note that a node may have two NICs with one NIC at each end of a network path. A network path may be included as a portion of another network path that communicatively couples a same pair of nodes. Data may be transmitted via the sequence of nodes in a network path between path end nodes communicatively coupled via the network path. Data may be transmitted in one or both directions depending on an ordering of the nodes in the sequence.

[0064] The term “hop” as used herein refers to a pair of consecutive nodes in a network path for transmitting, via a network protocol, data sent from a source node to a destination node. A “hop path” is thus a sequence of hops in a network that respectively include a sequence of pairs of consecutive nodes included in transmitting data from a first path end node of the network path to a second path end node of the network path.

[0065] The term “path-based protocol address” as used herein refers to a protocol address for a network protocol that includes a plurality of path segment identifiers that identify portions of a specific network path identified by the path-based protocol address. A “node-based protocol address” is a path-based protocol address that

includes a plurality of node identifiers that identify a sequence of nodes in a network path. A "network-interface-based protocol address" is a path-based protocol address that includes a plurality of network interface identifiers that identify a sequence of network interfaces in a network path. A "NIC-based protocol address" is a type of network-interface-based protocol address that includes a plurality of identifiers that identify a sequence of network interface components. A "hop-based protocol address" is a type path-based protocol address since a hop is a type of network path.

[0066] Given the above definitions, note that the terms "network path" and "hop" may be defined in terms of network interfaces. A "network path" and a "hop path" include a sequence of network interfaces in a network that are included in transmitting data between a pair of path end nodes in the network. A "hop" refers to at least part of a network path that includes a pair of consecutive network interfaces in a sequence of network interfaces in a network path. A "network path" is thus a sequence of hops in a network that respectively include a sequence of pairs of consecutive network interfaces included in transmitting data from a first path end node of the network path to a second path end node of the network path.

[0067] The term "network topology" as used herein refers to a representation of protocol endpoints and/or nodes in a network, and representations of hops representing communicative couplings between and/or among the protocol endpoints and/or nodes in the network. A network may have different network topologies with respect to different network protocols. A network topology may represent physical communicative couplings between nodes in the network. A network topology may represent logical couplings between protocol endpoints and/or nodes of a particular network protocol or a particular type of network protocol.

[0068] The domain name system (DNS) of the Internet operates based on an application layer protocol defined by the DNS. The nodes in the DNS are communicatively coupled via the DNS protocol and may be represented by a logical network topology. A DNS system includes nodes connected via the DNS protocol and thus has a network topology defined by nodes that include protocol endpoints of the DNS protocol. In still another example, a token-ring network has a circular topology at the link layer, but may have a star topology at the physical layer.

[0069] As used herein, an "entity-specific address space" refers to an address space defined for a specific entity where the addresses in the address space operate as identifiers in the context of the entity. An address from an entity-specific address space is referred to herein as an "entity-specific address". An address is "entity-specific" in that what it identifies is based on the entity to which it is specific. Another address having the same form and content may identify a different entity when in an address space specific to another entity. Addresses in an entity-specific address space operate as identifiers in the context of an entity to which they are "specific" as defined by the specific association of the address space and the entity. Without knowledge of the entity to which an entity-specific address space is specific, what an address in the entity-specific address space identifies is indeterminate. The terms "entity-specific address" and "entity-specific identifier" are used interchangeably herein. An entity-specific address may identify an entity included in the entity to which the address is specific or may identify an entity external to the entity to which the address is specific. The fact that an address is entity-specific does not define a scope for the address.

[0070] A portion of a network is a type of entity. A type of entity-specific address space described herein is a scope-specific address space. As used herein, a "scope-specific address space", specific to a particular region of a network, is an address space defined for the particular network region, where an address in the scope-specific protocol address operates as identifier, according to a network protocol, of a protocol endpoint in a node outside of the particular region when processed in the context of a node in the particular region. The region is indicated by the span of an indicated scope. The terms "region" and "zone" are used interchangeably herein. An address from a scope-specific address space is referred to herein as a "scope-specific protocol address". An address is "scope-specific" in that what protocol endpoint it identifies depends on the region to which it is specific. Another address having the exact same form and content may identify a different protocol endpoint when in an address space that is specific to another region. A protocol address in a scope-specific address space serves as an identifier in the context of a node in a region to which the scope-specific address space is "specific" as defined by an association of the address space and the region indicated by the scope. Without knowledge of the particular region to which a scope-specific

address space is specific, what a scope-specific protocol address in the scope-specific address space identifies is indeterminate. The terms “scope-specific protocol address” and “scope-specific protocol identifier” are used interchangeably herein. Types of scope-specific address spaces indicating exemplary spans include site-specific, LAN-specific, subnet-specific, city-specific, business-specific, and node-specific.

[0071] For a network protocol, an address in a scope-specific address space serves as an identifier of a protocol endpoint in a node. Data may be received via the protocol endpoint from a network via one or more network interfaces that operatively couple the node to the network. Data may be sent via the protocol endpoint for transmitting over the network via the one or more network interfaces in the node. Since a protocol endpoint of a network protocol is included in a node and is accessible via a network via a network interface, a protocol address identifying the protocol endpoint also identifies the node and identifies a network interface of the node.

[0072] As used herein, a “node-specific address space” is a scope-specific address space defined for a specific node in a network, where the addresses in the node-specific address space operate as identifiers of nodes and/or network interfaces in the network when processed in the context of the specific node. An address from a node-specific address space is referred to herein as a “node-specific address”. An address is “node-specific” in that what it identifies depends on the node to which is defined as specific. Another address having the exact same form and content may identify a different node when in an address space specific to another node. Addresses in a node-specific address space operate as identifiers in the context of a node to which they are “specific” as defined by the specific association of the address space and the node. Without knowledge of the node to which a node-specific address space is specific, addresses in the node-specific address space are indeterminate. The terms “node-specific address” and “node-specific identifier” are used interchangeably herein. A node-specific address space is a type of scope-specific address space.

[0073] The term “node” is defined above. Note that an identifier of a network interface in a network also identifies a node that includes the network interface. Thus, a network interface-specific address is also a node-specific address. Network interfaces in a node may have their own respective network interface-specific address spaces that are also

node-specific. The network interface-specific address spaces may be combined to form a node-specific address space and/or may be managed as separate address spaces. The adjectives “node-specific” and “network interface-specific” may be used interchangeably.

[0074] A scope-specific identifier differs from a scoped address as described in "Request for Comments" (RFC) document RFC 4007 by S. Deering, et al, titled "IPv6 Scoped Address Architecture”, published by the IETF in December, 2006 and further described in Application No. 11/962,285, by the present inventor, filed on 2007/12/21, entitled “Methods and Systems for Sending Information to a Zone Included in an Internet Network”. A scoped address space is shared by nodes in a given scope. While a link-local scoped address is specific to a particular node, a link-local scoped address simply identifies a network interface component local to the particular node. A loop-back internet address is specific to a node as well. Neither link-local scoped addresses nor loop-back addresses identify one node to another. As such, neither serves as a node-specific identifier as defined above.

[0075] A “scoped address” is described by RFC 3513 and RFC 4007 as an identifier that, in a particular region of a network, serves as a protocol address of a network interface and/or a node in the particular region. The extent of the particular region is referred to as the scope of the region and thus the scope within which the identifier serves as a protocol address. A particular region included within a scope as indicated by its span. A scoped address is a valid protocol address only within a particular region as indicated by the address’s indicated scope. Examples of scope indicators include node-scope where identifiers are valid only to a single node in the indicated span, LAN-scope where identifiers are valid for nodes in the span of a particular LAN, and subnet-scope where identifiers are valid only for nodes in a particular subnet. RFC 3513 currently defines support for link-local scope, site-local scope, a global scope. A data unit transmitted with a scoped address should not be delivered to node that does not have a network interface in the span indicated by the scope.

[0076] “Path information” is any information that identifies a network path and/or hop path for data transmitted via one or more specified network protocols. Path information may be identified by identifying network interfaces, NICs, nodes, and/or hops included

in a network path. “Address information” is any information that identifies a protocol address that, for a network protocol, identifies a protocol endpoint. Address information may identify a unicast protocol address for a network protocol. In identifying a protocol endpoint, a protocol address identifies a node and a network interface. Those skilled in the art will understand upon reading the descriptions herein that the subject matter disclosed herein is not restricted to the network protocols described and/or their corresponding OSI layers. For ease of illustration, the subject matter is described in terms of protocols that correspond to OSI layer three, also referred to as network layer protocols, in general. Particular descriptions are based on versions of the Internet Protocol (IP). Address information may identify one or more protocol addresses. Exemplary protocol addresses include IP addresses, IPX addresses, DECNet addresses, VINES Internet Protocol addresses, and Datagram Delivery Protocol (DDP) addresses.

[0077] The term “path-based address” is defined above. A “node-based address” is a path-based address where some or all of the address includes node identifiers that identify a sequence of nodes in a network path. A “network-interface-based address” is a path-based address where some or all of the address identifies includes identifiers of a network interfaces in sequence in a network path. A “NIC-based address” is a type of network-interface-based address that identifies a sequence of network interface components. A “hop-based address” is a path-based address where some or all of the address identifies one or more hops in a network path. The protocol address types defined are not mutually exclusive.

[0078] The term “metric space”, as used herein, refers to a set, as defined in mathematics, where a distance between elements of the set is defined according to a metric. Metric spaces defined in Euclidean geometry are well-known examples. Those skilled in the art of metric spaces, such as Euclidian spaces, will appreciate that a one-to-one mapping may be determined and/or otherwise identified for mapping addresses from a first coordinate space having a first origin for a metric space to addresses from a second coordinate space having a second origin in the metric space. Given a mapping rule between a first scope-specific address space and a second scope-specific address space and a mapping between the second scope-specific address space and a third

scope-specific address space based on a third coordinate space identifying a third origin in the metric space, a mapping from the first coordinate space to the third coordinate space may be determined. A mapping between coordinate spaces for a metric space may be included a coordinate shift and/or a rotation, for example. The mapping may be pre-specified and accessible to the nodes in one or both address spaces. Mapping between locations in a number of different metric spaces is well-known in mathematics. For example, a top half of the surface of sphere may be mapped to a plane. Some will further appreciate that some metric spaces may be mapped to other metric spaces. Some of these mappings are one-to-one and onto.

[0079] An "interaction", as the term is used herein, refers to any activity including a user and an object where the object is a source of sensory data detected by the user. In an interaction the user directs attention to the object. An interaction may also include the object as a target of input from the user. The input from the user may be provided intentionally or unintentionally by the user. For example, a rock being held in the hand of a user is a target of input, both tactile and energy input, from the user. A portable electronic device is a type of object. In another example, a user looking at a portable electronic device is receiving sensory data from the portable electronic device whether the device is presenting an output via an output device or not. The user manipulating an input component of the portable electronic device exemplifies the device, as an input target, receiving input from the user. Note that the user in providing input is detecting sensory information from the portable electronic device provided that the user directs sufficient attention to be aware of the sensory information and provided that no disabilities prevent the user from processing the sensory information. An interaction may include an input from the user that is detected and/or otherwise sensed by the device. An interaction may include sensory information that is detected by a user included in the interaction that is presented by an output device included in the interaction.

[0080] As used herein "interaction information" refers to any information that identifies an interaction and/or otherwise provides data about an interaction between a user and an object, such as a portable electronic device. Exemplary interaction information may identify a user input for the object, a user-detectable output presented by an output

device of the object, a user-detectable attribute of the object, an operation performed by the object in response to a user, an operation performed by the object to present and/or otherwise produce a user-detectable output, and/or a measure of interaction. The term "operational component" of a device, as used herein, refers to a component included in performing an operation by the device.

[0081] Interaction information for one object may include and/or otherwise identify interaction information for another object. For example, a motion detector may detect user's head turn in the direction of a display of a portable electronic device. Interaction information identifying the user's head is facing the display may be received and/or used as interaction information for the portable electronic device indicating the user is receiving visual input from the display. The interaction information may serve to indicate a lack of user interaction with one or more other objects in directions from the user different than the detected direction, such as a person approaching the user from behind the user. Thus the interaction information may serve as interaction information for one or more different objects.

[0082] The term "attention information" as used herein refers to information that identifies an attention output and/or that includes an indication to present an attention output. Attention information may identify and/or may include presentation information that includes a representation of an attention output, in one aspect. In another aspect, attention output may include a request and/or one or more instructions for processing by a processor to present an attention output. The aspects described serve merely as examples based on the definition of attention information, and do not provide an exhaustive list of suitable forms and content of attention information.

[0083] As used herein the term "attention criterion" refers to a criterion that when met is defined as indicating that interaction between a user and an object is or maybe inadequate at a particular time and/or during a particular time period. In other words, the user is not directing adequate attention to the object.

[0084] As used herein, the term "tag" refers to a character string, which may include one or more words, which may be associated with a resource to create an association between the resource and another resource also tagged with the tag. Tags are often used in sharing media, social bookmarking, social news and blog entries to help users

search for associated content. In some contexts, the term “tagging” as used herein refers to the process of associating a tag with a resource that can be tagged. As used herein, the term “tagged association” refers to an association that identifies a tag and a resource that is tagged with the tag. A resource is said to be “tagged” with a tag when a tagging process, also referred to as “tagging”, has successfully created a tagged association. In other contexts, a “tagging” refers to a tagged association.

[0085] The term “matched tag”, as used herein, refers to an association that identifies a tag and a tagging criterion, where the tagging criterion is based on data other than the tag. The tagging criterion may be based additionally on the tag. The term “tagging association”, as used herein, refers to a matched tag represented in a memory, either persistent and/or volatile. For example, a tagging association may be realized as a record that includes and/or identifies a tag and a tagging criterion. The terms “matched tag” and “tagging association” are used interchangeably. When the term “matched tag” refers to something other than its tagging association, it will be made clear. For example, a reference to matched tag presented in a user interface clearly identifies a UI element that identifies and/or otherwise represents and/or otherwise identifies a matched tag either realized or to be realized in a tagging association. A tagged association, defined above, may identify a matched tag and a resource that is tagged with the tag identified by the matched tag. Such an association is also referred to herein as a “matched tag association”. A resource may be tagged with a matched tag when a tagging criterion identified by the matched tag is met for the resource.

[0086] As user herein, the term “vocabulary” refers to a set of valid values that can be assigned to and/or included in a data element. With respect to a tag, a vocabulary defines valid tags. A vocabulary may be specified by one or more rules and/or by identifying one or more valid values directly and/or indirectly.

[0087] As used herein, the term” dictionary” refers a vocabulary wherein one or more terms in the vocabulary is assigned a definition. A “definition” as the term is used herein refers to information that identifies semantic information about a data value, such as a t tag. Semantic information may include one or more of a textual description of a meaning of the term, an audio description, a visual description, and information identifying a use or context for the term. A use and/or context may be identified, for example, by

identifying a part of speech, a dialect or language, an antonym, a synonym, and/or an example usage of the defined data value.

[0088] As used herein, any reference to an entity "in" an association is equivalent to describing the object as "identified" by the association, unless explicitly indicated otherwise.

[0089] As used herein, the term "communication" refers to information including a message sent and/or for sending via a network between communicants. A message may include text data, audio data, and/or image data. The term "communicant" as used herein refers to a user included in a communication as a sender and/or a receiver of the information. A communicant is represented by a "communications agent" configured to operate in an execution environment to send data to and/or receive data from another communications agent, on behalf of the represented communicant, according to a communications protocol via network. A communications protocol defines and/or otherwise identifies an address space including communications addresses for delivering data sent in a communication from one communications agent to another. A communications protocol is a type of network protocol.

[0090] The term "communicant alias" as used herein refers to an identifier of a communicant in a communication where the communicant alias is not a communications address included in an address space of a communications protocol for sending and/or receiving data in the communication.

[0091] The term "attachment" as used herein refers to a portion of a communication that includes data from one communicant to another other than data in the message portion. A resource sent as an attachment is data that is typically not presented "inline" or in a message included in a message portion of a communication. Email attachments are perhaps the most widely known attachments included in communications. An email attachment is a file or other resource sent along with an email in a portion of the email separate from a message portion. A communication may include one or more resources as one or more attachments.

[0092] The terms "contactor" and "contactee" identify roles in a communication. A "contactor" provides information for identifying a "contactee" in a communication. A contactee may be included in a communication by a contactor and/or the contactor's

communications agent. A contactor and/or a contactee are roles of a user or communicant of a communications agent and may played by a living being, a node, a component, and/or an application. Both a contactor and a contactee are communicants in a communication.

[0093] As used here, the term "tagging- attribute association" refers to an association that identifies one or more tagging criterion and a resource attribute that the one or more tagging criterion are each based on. That is, whether a tagging criterion identified in a tagging-attribute association is met for a specified resource depends on an attribute of the resource identified by the tagging-attribute association. The term attribute, as used herein, includes the resource, as a resource is considered to be an attribute of itself and/or otherwise to include at least some of its attributes.

[0094] As used herein, the term "resource scope" refers a bounded space in which a resource may be identified. The space may be any space in which a resource may be locate and/or otherwise identified. A bounded may be may include one or more of an identifier space such as file system identifier space and/or a URI address space, a storage space, a geospatial space, a demographic space, and a data type – to name a few examples. space, defines a set of resources from which resources may be identified. For example, the files within a specified folder and its subfolders in a file system define and/or otherwise specify a resource scope. All user accounts that have an administrator security role define another resource scope. The present disclosure describes a resource scope with respect to resources tagged with a specified tag that makeup a set, referred to herein as a "tagged resource set". A tagged resource set may include a resource tagged with a tag identified by a matched tag.

[0095] In specifying a resource scope, one or more criterion may be specified. The terms "criterion" and "constraint" are used interchangeably herein, with respect to a resource scope. A criterion included in specifying a resource scope is referred to herein as a "scope criterion". A scope criterion may be expressed via any suitable manner, such as a file name, a wildcard expression, a regular expression, a data store location, a data base query, geospatial coordinates, a time, a date, a duration, a relationship to a user, and/or an address – too name a few examples. With respect to a tag, a scope criterion included in defining a resource scope differs from a tagging criterion for the tag

in that the scope criterion included in defining a resource scope need not be met in order for a resource to be tagged with the tag. Further a resource scope may be used for purposes other than processing related to a tagged resource.

[0096] One or more scope criterion that together specify and/or otherwise define a resource scope is/are referred to herein as a "scope descriptor". When all the criteria in a scope descriptor are met for a resource, the resource is in the resource scope defined by the scope descriptor. The one or more separate criterion in a scope descriptor may be evaluated in any order suitable or desirable in any given embodiment. In an aspect described below, one scope criterion, that may be included in a scope descriptor, may be specified to determine and/or otherwise identify when the scope descriptor is "active". A scope criterion that indicates whether a scope descriptor is active is referred to herein as a "scope condition". In a further aspect a scope descriptor may include no other criterion indicating the resource scope includes all resources accessible to an application, subsystem, execution environment, and/or group of any of the preceding that are processing and/or otherwise making use of the scope descriptor. Those skilled in the art will understand that whether a scope criterion in a scope descriptor is called a scope condition or not depends on the role played by the scope criterion. The subject matter may be described without referring to a scope criterion in the role of a scope condition without limiting the scope of subject matter.

[0097] Although, those skilled in the art will know that a scope criterion may be defined as "met" when it is true or may be defined as "met" when it is not true; for purposes of the present disclosure, A scope criterion is said to be "met" for a resource when it indicates the resource is in the resource scope defined by the scope descriptor, and is to be "not met" otherwise. A scope criterion in the role of a scope condition in a scope descriptor is said to be "met" when it indicates the scope descriptor is active, as is said to be "not met" otherwise. In an aspect, a scope descriptor may not include a scope condition or may include a scope condition that is always met. Such a scope descriptor is active by definition. Whether a criterion, in a scope descriptor is met or is not met, may depend on an evaluation of the scope criterion that is based on an identifiable entity. For a scope condition, such an entity is referred to herein as a "condition entity". More generally, for a scope criterion, such an entity is referred to herein as a "scoping

entity". Exemplary entities include a user, a document, a measure of time, a location, and the like.

[0098] An association between a tag and scope descriptor is referred to herein as a "scoped tag". A scoped tag is also referred to herein as a match set. As defined above, a tagging criterion may be used to determine whether to tag a particular resource with a particular tag. The tagging criterion is associated with the tag by a matched tag. A scope descriptor identifies resource scope that includes a set of resources. Scope descriptors are necessarily associated with resources.

[0099] A "tagging context", as the term is used herein, identifies and/or otherwise includes a set of tagged associations or taggings. The taggings may identify one or more tags and one or more resources. A "tagging context specification", as the term is used herein, is data that defines and/or otherwise specifies a tagging context. A tagging context specification may be stored in a memory location. The memory location may be contiguous or not and may be included in one or more data storage media, which may be volatile or not. A tagging context specification may include and/or otherwise identify any suitable criterion, referred to in general as "context criterion", for determining whether a tagging is in a tagging context defined by the tagging context specification. For example, a tag contact specification may identify one or more scope descriptors, one or more tags some or all of which may be and/or may be identified by matched tags, and a context condition; all of which are context criterion by definition.

[0100] A "context condition", as the term is used herein, refers to a context criterion that when met indicates that a tagging context is active for one or more taggings. When a context condition is not met, it is said to be inactive or not active for one or more taggings. For example, a context condition in a tagging context specification may be met for a first user. Taggings in the tagging context are "active" for the first user. For a second user, the context condition may not be met. The taggings are not active for the second user. A context condition, or more generally a context criterion, may be based on any suitable data for any suitable entity. A tagging context may be active for a task and not for another and/or a tagging context may be active for one location and not for another, to name a couple of other examples. When a context condition indicates that

an associated tagging context is active for a tag, the context condition is herein said to be "met", otherwise the context condition is not met.

[0101] FIG. **3A** illustrates an arrangement of components in a system that operates in an execution environment, such as execution environment **102** in FIG. **1**. The arrangement of components in the system operates to perform the method illustrated in FIG. **2A**. The system illustrated includes a query handler component **302**, a identification request constructor (IRC) component **304**, and a result handler component **306**. A suitable execution environment includes a processor, such as processor **104**, to process an instruction in at least one of a query handler component, a identification request constructor component, and a result handler component. FIG. **3B** illustrates an arrangement of components in a system that operates to perform the method illustrated in FIG. **2B**. The system illustrated includes a content manager component **309**, a query handler component **302**, a query director component **312**, and a identification response builder (IRB) component **314**. A suitable execution environment includes a processor, such as processor **104**, to process an instruction in at least one of a content manager component, a query handler component, a query director component, and a identification response builder (IRB) component.

[0100] Some components, illustrated in the drawings are identified by numbers with an alphanumeric suffix. A component may be referred to generically in the singular or the plural by dropping a suffix of a portion thereof of the component's identifier. For example, window UI elements, such as a window UI element **602a** in FIG. **6A**, a window UI element **602b** in FIG. **6B**, and their adaptations and analogs; are referred to herein generically as a window UI element **602** or window UI elements **602** when describing more than one. Other components identified with an alphanumeric suffix may be referred to generically or as a group in a similar manner.

[0101] The arrangement components illustrated in FIG. **4** may be adapted to operate in a number of execution environments to perform the method illustrated in FIG. **2A** and/or the method illustrated in FIG. **2B**. FIG. **4** is a block diagram illustrating the components of FIG. **3A**, FIG. **3B**, and/or analogs of the components of FIG. **3A** and FIG. **3B**, respectively adapted to operate in an execution environment **401** that includes and/or otherwise is provided by one or more nodes. FIG. **1** illustrates key components of an

exemplary device that may at least partially provide and/or otherwise be included in an execution environment. The components illustrated in FIG. **4** may be included in or otherwise combined with the components of FIG. **1** to create a variety of arrangements of components according to the subject matter described herein.

[0102] As stated, the various adaptations of the arrangement in FIG. **3A** as well as the various adaptations of the arrangement in FIG. **3B** illustrated and described herein are not exhaustive. For example, those skilled in the art will see based on the description herein that arrangements of components for performing the method illustrated in FIG. **2A** and the method illustrated in FIG. **2B** may each be distributed across more than one node and/or execution environment. For example, such an arrangements may operate at least partially in a browser in a one node and at least partially in a server in another node interoperating via a network.

[0103] FIG. **5** illustrates a first node **502** and an execution environment **401** of second node **504** as exemplary devices that each may be included in and/or otherwise adapted for providing an instance, adaptation, and/or analog of execution environment **401** in FIG. **4**. As illustrated in FIG. **5**, first node **502** and an execution environment **401** of second node **504** are operatively coupled to a network **506** via respective network interface components enabling first node **502** and execution environment **401** of second node **504** to exchange data in a communication on behalf of communicants represented by the respective nodes.

[0104] FIG. **4** illustrates communications agent **403** including an adaptation of the arrangement of components in FIG. **3A** and an adaptation of the arrangement of components in FIG. **3B**. A first communications agent **403** may operate, in first node **502**, on behalf of a first communicant to communicate with a second communicant represented by a second communications agent **403** operating in execution environment **401** of second node **504**.

[0105] Execution environment **401** is illustrated in FIG. **4** hosting communications agent **403**. Components in FIG. **4** are referenced for illustrative purposes in describing communications agents operating in an execution environment of first node **502** and an execution environment of execution environment **401** of second node **504**. Exemplary communications agents include email clients, phone clients including Voice over

Internet Protocol (VoIP) clients, instant messaging clients, short message service (SMS) clients, multimedia message service (MMS clients), multi-media communications clients including video phone clients, and other data transfer agents.

[0106] Communications agents **403** in FIG. **4**, respectively operating in first node **502** and in execution environment **401** of second node **504** in FIG. **5** may interoperate via respective network stacks **405**. Communications agents **403** may communicate via one or more communications protocols. FIG. **4** illustrates communications protocol component **407** exemplifying a subsystem for exchanging data via network **506** according to one or more communications protocols, such as simple mail transfer protocol (SMTP), an instant messaging protocol, and/or a real-time voice and/or video protocol. A communication between communications agents **403** in first node **502** and execution environment **401** of second node **504** may include more than one type of data and may use one or more communications protocols in exchanging one or more types of data via network **506**.

[0107] Instances, adaptations, and/or analogs of communications agent **403** in FIG. **4**, in first node **502** and in execution environment **401** of second node **504**, respectively, may communicate via discrete messages, a request/reply protocol, a data streaming protocol, a session and/or connection-oriented protocol, a connectionless protocol, a real-time communications protocol, an asynchronous communication, a store and forward communications protocol, a reliable delivery communications protocol, a best-effort delivery communications protocol, and/or a secure protocol, to name a few communications options.

[0108] FIG. **4** illustrates communications agent **403** including a content manager component **409**. Content manager component **409** may interoperate with communications protocol layer component **407** and/or network stack **405** to receive data in one or more communications via network **506** with another communications agent in another node. Content manager component **409** is operatively coupled, via com-in component **411**, to communications protocol component **407** for receiving the data from the other node.

[0109] Data received in a communication may include one or more resources and/or content types. Exemplary content types include plain text, markup such as hypertext

markup language (HTML), audio data, image data, and/or executable data. Executable data may include script instruction(s), byte code, and/or machine code. In FIG. **4**, communications agent **403** includes one or more content handler components **413** to process data received according to its content type. A data type may be identified by a MIME type identifier. Exemplary content handler components **413** include a text/html content handler component for processing HTML representations; an application/xmpp-xml content handler component for processing extensible messaging and presence protocol (XMPP) streams including presence tuples, instant messages, and audio content handlers including and/or configured to retrieve suitable codices; one or more video content handler components for processing video representations of various types; and still image data content handler components for processing various image data representations.

[0110] Content handler component(s) **413** process received data representations and may provide transformed data from the representations to one or more user interface element handler components **415**. One or more user interface element handler components **415** are illustrated in a presentation controller component **417** in FIG. **4**. Presentation controller component **417** may manage visual, audio, and other types of output for its including application as well as receive and route detected user and other inputs to components and extensions of its including application, communications agent **409**. A user interface element handler component **415** may be adapted to operate at least partially in a content handler component **413** such as a text/html content handler component and/or a script content handler component. Additionally or alternatively, a user interface element handler component in an execution environment **401** may be received in a communication. For example, a communication, such as an email, may include an HTML content type portion and a script content type portion.

[0111] FIG. **6A-G** illustrates various communications agent windows **602** presentable in a presentation space of a display device, such as output device **130** in FIG. **1**. In FIG. **6a**, a communications agent window **602a** includes a contactor user interface (UI) element **604a** for presenting an identifier of a communicant in the role of a contactor in a communication represented by the communications agent window **602a**. A communications agent window **602a** also includes a contactee UI element **606** for

presenting one or more contactee identifier(s) identifying one or more communicants in the role of contactee(s) included in the communication. A presentation space **608** is provided by each communications agent window **602a** for presenting a user message UI element **610** for presenting a message addressed to one or more contactees identified in a contactee UI element **606a**. A presentation space **608a** may also be provided for presenting one or more UI controls for exchanging data in and/or otherwise managing a communication. A send UI element **612a** illustrates an exemplary UI element that may correspond to user input to send data in a communication to one or more identified contactees.

[0112] Data to send in a communication to a communications agent in execution environment **401** of second node **504** may be received by one or more content handler component(s) **413** operating in first node **502** to transform the data into one or more data representations suitable for transmitting in the communication and/or suitable for processing by the communications agent in execution environment **401** of second node **504**. The one or more data representations may be provided to content manager component **409** for sending in the communication to execution environment **401** of second node **504**. Content manager component **409** may package the one or more data representations in a message formatted according to a communications protocol of the communications agent. Communications protocol component **407** may send the data according to the specification(s) of the communications protocol. Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in the communication to a communications agent via a network.

[0113] Content manager component **409** operating in execution environment **401** included in and/or otherwise provided by first node **502** may provide the packaged, encoded, and/or transformed data to communications protocol component **407** via a com-out component **419**. Com-out component **419** as described above operatively couples communications agent **409** to communications protocol component **407** according to an interface provided by communications protocol component **407** for sending data in a communication according to a communications protocol.

Communications protocol component **407** may further package and/or otherwise transform the data to send via network stack **405** for delivery via network **506** to execution environment **401** of second node **504**.

[0114] As described above, a communications protocol may operate via one or more nodes in a network in a network path including a first node and an execution environment **401** of second node **504**. Exemplary path nodes include mail relay nodes, phone switch nodes, and proxy nodes such as instant messaging proxies for communicating through firewalls. As indicated, path node **508** illustrates such a node.

[0115] Sending data in a communication requires identifying a contactee. A contactee may be represented by a communications address in an address space of a communications protocol. In one aspect, information identifying a communications address may be received from a user/communicant of a communications agent in a user node. In FIG. **4**, presentation controller **417** and/or a UI element handler **415**, presenting and/or managing interaction with contactor UI element **604** in FIG. **6A** for first node **502**, may receive a contactor alias in response to a user input corresponding to UI element handler **604**. The user/communicant of first node **502** may enter a contactor alias, such as “Dad”, via a keyboard and/or select a predefined communicant alias presented in a selection UI control element via a UI element handler component **415**. The user input may be detected by input driver **421**. Corresponding input information may be routed to presentation controller **417** by GUI subsystem **423**. GUI subsystem **423** may send presentation information to a display device via a graphics subsystem **425**. Communications agent **403** may identify a communication address associated with the contactor alias, “Dad”. Other communications address(es), such as for one or more contactees, may be received similarly and/or in any suitable manner.

[0116] Data may be sent in a communication according to a form or type of the communication and/or other attribute of the communication such as a security attribute, the amount of data to be set, a priority setting, a task setting, and the like. Some forms of communication do not require a session and/or connection between a first node and an execution environment **401** of second node **504** in order to send data between the two nodes, while others do. An email and/or instant message may use a store and forward model of delivery.

[0117] Data may be sent in a communication in response to a communicant input. A contactor may provide an input corresponding to send UI element **612a** in FIG. **6A**. The input may be received by presentation controller component **417** and/or one or more UI element handlers **415** corresponding to send UI element **612a**. In response to detecting the input, presentation controller component **417** may provide data to be sent in the communication to one or more content handler components **413** according to the content type(s) of the data to be sent.

[0118] The one or more content handler components **413** may encode, format, and/or otherwise transform the data for sending in a message, such as an email message. The one or more content handler components **413** may provide data to be sent to content manager **409**, instructing content manager component **409** to send the data in the communication for delivery to a communications agent in execution environment **401** of second node **504**. Content manager component **409** interoperating with com-out component **419** may further format and/or transform the data for sending in the communication according to communications protocol, for example according to an email communications protocol, by communications protocol component **407**. Communications protocol component **407** may send the communication for delivery to the communications agent in execution environment **401** of second node **504** via network **506**.

[0119] Path node **508** may relay data sent in the communication between first node **502** and execution environment **401** of second node **504**. Path node **508** may determine a next node and/or a network interface in a network path communicatively coupling first node **502** and execution environment **401** of second node **504** for exchanging data in a communication between communications agents **403** in the first node **502** and in the execution environment **401** of second node **504**.

[0120] For session-oriented and/or connection-oriented communication a session and/or connection may be established if a session/connection has not already been established. Data may be sent for delivery to a communications agent identified based on a contactee communications address during session and/or connection setup. For example, for a voice communication a voice communication may be established via a session initiation protocol. Communications protocol component **407** may operate

according to the session initiation protocol specifications. Communications protocol component **407** operating in first node **502** may locate a communications agent by communicating with one or more nodes in network **506** according to the session initiation protocol. Communications protocol component **407** may locate a communications agent in execution environment **401** of second node **504**, based on a communications address for the contactee located based on the contactor alias.

[0121] Once a communication session is established, such as a voice session, data may be sent according to the session communications protocol, such as RTP. Data may be sent according to the session initiation protocol in the communication for managing the voice communication session and/or for exchanging text, image, and/or other data outside of the voice session. Path node **508** may be included in session and/or connection setup. Alternatively or additionally, path node **508** may be included in a network path in a session and/or connection.

[0122] With reference to FIG. **2A**, a block **202** illustrates that the method includes receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. Accordingly, a system for processing a data object identification request in a communication includes means for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. For example, the arrangement in FIG. **3A**, includes query handler component **302** that is operable for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. FIG. **4** illustrates query handler component **402** as an adaptation and/or analog of the query handler component **302** in FIG. **3A**. One or more query handler components **402** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object

information identifying a data object matching criterion. In FIG. **4**, query handler component **402** is illustrated as a component of communications agent **403**.

[0123] FIG. **7** includes dataflow message **702** that illustrates a data flow internal to execution environment **401** of first node **502** and/or a message received via network **506** by execution environment **401** of first node **502**. Dataflow message **702** illustrates a data object information dataflow to query handler component **402** operating in execution environment **401** of first node **502**. Data object information is identified to and/or otherwise received by query handler component **402**.

[0124] In various aspects, a query handler component **402** may receive data object information from a user via one or more UI element handler components **415**. The data object information may be received, in response to a user input detected by an input device of an execution environment **401** that includes the query handler component **402**. Alternatively or additionally, data object information may be received by a query handler component in a message received via a network. For example, a user may identify data object information to a browser operating in a client node. The browser may send the data object information via a network communicatively coupling the client node to a web service provider node. The web service provider node may include a query handler component that may operate to receive the data object information identified in the message from the browser. FIG. **6A** illustrates an exemplary user interface that may be presented by communications agent **403** operating in execution environment **401** of first node **502**. A UI element handler component **415** may operate to present a user interface element to allow a user to identify data object information. A data object information (DOI) UI element **616a** is illustrated in FIG. **6A** as an example. The same or different UI element handler component **415** may present various other UI elements included in DOI UI element **616a**. DOI UI element **616a** and one or more UI elements it includes may be presented by one or more corresponding UI element handler component(s) **415** that may operate based on a schema that defines valid data object information and/or data object identification requests for including in a communication via a communications agent **403**. The schema defines one or more rules and/or a vocabulary that defines whether data object information and/or a data object identification request is valid. Those skilled in the art will understand that

numerous languages and/or schemas for providing data object information and/or data object identification requests currently exist including relatively simple key word based queries to relatively more complicated regular expression languages and data base query languages, such as various versions of structured query language (SQL). New schemas are and will be created, and are, thus, considered within the scope of the subject matter described herein.

[0125] DOI UI element **616a** illustrates a user interface for a keyword based schema. Match textbox UI element **618a** illustrates a textbox allowing a user to enter a keyword expression. Various checkbox UI elements **620a** allow a user to define a scope of a data object identification request by identifying whether a resulting data object identification request is to be applied to folders, files, and/or content of files in a file system and/or other data store of a receiving execution environment. A communications agent **403** may support more than one data object identification request schema and/or corresponding data object information schema. Thus. a communications agent may provide a user interface to receive valid data object information for each respective schema.

[0126] Returning to FIG. **2A**, a block **204** illustrates that the method further includes sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. Accordingly, a system for processing a data object identification request in a communication includes means for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. For example, the arrangement in FIG. **3A**, includes identification request constructor component 304 that is operable for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the

data object matching criterion, wherein the first message is addressed to the second user. FIG. **4** illustrates identification request constructor component **404** as an adaptation and/or analog of identification request constructor component **304** in FIG. **3A**. One or more identification request constructor components **404** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. In FIG. **4**, identification request constructor component **404** is illustrated as a component of communications agent **403**.

[0127] FIG. **7** includes dataflow message **704** that illustrates a data flow internal to execution environment **401** of first node **502** and/or message received via network **506** by execution environment **401** of first node **502**. Dataflow message **704** illustrates a create data object identification request (CIReq) dataflow to an IRC component **404** operating in execution environment **401** of first node **502**. Data object information may be identified to and/or otherwise provided to IRC component **404**. A query handler component **402**, operating as described with respect to block **202** of FIG. **2A**, may identify and/or otherwise provide data object information to an IRC component **404**.

[0128] A query handler component may be a type of content handler component that operates to process data object information to create a data object identification request that conforms to a schema for creating and/or otherwise constructing a valid data object identification request. The IRC component **404** may construct and/or otherwise create a data object identification request based on a schema that defines and/or otherwise identifies a valid data object identification request and/or a valid data object identification response for a particular type of communication supported by a communications agent. The data object identification response is constructed and/or otherwise created, based on the data object information, by the IRC component **404**. IRC component **404** in execution environment **401** of first node **502** may provide the

data object identification request to content manager component **409** to include and/or otherwise identify the data object identification request in a communication with execution environment **401** of second node **504**, such as email and/or instant message. Content manager component **409** may interoperate with other types of content handler components **413** to create and/or otherwise construct a message that includes the valid data object identification request.

[0129] In an aspect, the IRC component **404** may interoperate with the content manager component **409** to create the valid data object identification request according to a specified schema for the message, so that the message is valid according to a communications protocol supported by a communications protocol component **407** that operates to send the message. As described above, the content manager component **409** may operate in an execution environment **401** of first node **502** along with an IRC component **404** to transform data object information into a data object identification request to include along with data for other parts of a communication into one or more representations suitable for transmitting in a communication to another node, such as execution environment **401** of second node **504**. Some or all of the representations transmitted are suitable for processing by the communications agent in execution environment **401** of second node **504**. The content manager component **409** in the execution environment **401** of first node **502** may package the one or more data representations including a representation of the data object identification request into a message formatted according to the communications protocol.

[0130] In FIG. **8A**, a portion of an email communication **800a** is illustrated formatted as a multipart/mixed content type including data object identification request portion **802a**. A data object identification request portion of a communication may be identified as a data object identification request by its location in the communication and/or by an identifier or markup element, such as a MIME type identifier. A data object identification request may be detected based on content included in the message and/or based on metadata such as content-type header **804a** identifying a MIME type identifier, such as “application/id-request”, which may be defined for representing one or more matching criteria in a data object identification request. The “application/id-request” MIME type

identifier is exemplary. Other MIME type identifiers exist and/or may be defined to identify a data object identification request in a communication.

[0131] An IRC **404** and/or a content manager component **409** may operate to construct a data object identification request in a communication based on XQuery, regular expression, and/or SQL content – to name a few examples. A content type identifier may be included in a position and/or location that identifies a data object identification request in a communication. The position or location may be absolute or relative. For example, a schema for a communication may define that a data object identification request in a communication is included in the communication at the end of the communication. There may be one or more data object identification requests at the end. In another aspect, a schema for a communication may specify that a portion of a communication following a particular type of message portion is a data object identification request. Other data object identification requests may follow. If no data object identification request is included, the data object identification request portion may include no content or may include an indicator that no data object identification request is included.

[0132] FIG. **8A** illustrates an "application/id-request" MIME type identifier that may be defined to identify a schema for an XML-based language for specifying id-request XML documents. FIG. **8A** illustrates id-request document **806a**. Id-request document **806a**, as illustrated, includes criterion tag elements **808a** corresponding to the form elements in the user interface illustrated in FIG. **6A**. A criterion tag element **808a** identifies a data object matching criterion name, such as "query" indicating that the criterion is a query expression or a portion thereof. A "type" attribute identifies a schema with the identifier "keyword" for the expression. The criterion tag element **808a** identifies a value for a query "Uncle Joe" in a match-expression attribute. Another criterion tag element **808a** specifies a scope query specified in the first criterion tag element. FIG. **8A** illustrates an "and" tag **810a** indicating that all the matching criteria must be met in identifying a data object. An "or" tag (not shown) may be defined by a schema for id-request documents. Other operator elements and operator precedence may be defined by the schema. Grouping elements for managing operator precedence, such as a parenthesis element, may be defined by the schema.

[0133] FIG. **7** includes dataflow message **706** that illustrates a first message, including a data object identification request, sent via network **506** by execution environment **401** of first node **502** to execution environment **401** of second node **504**.

[0134] A data object identification request generated by an IRC component **404**, along with a user message and any other data to include in a communication, may be provided and/or otherwise identified to content manager component **409** for sending in the communication. The content manager component **409** in execution environment **401** of first node **502** may package the data including a representation of the data object identification request into the message formatted according to the communications protocol. Com-out component **419** may provide the data object identification request, the user message, and any other data for sending in the communication in representations suitable for sending by communications protocol component **407** to the communications agent in execution environment **401** of second node **504**. Communications protocol component **407** operating in execution environment **401** of first node **502** may send the data as and/or in a communication, such as an email message, according to the specification(s) of the communications protocol. Communications protocol component **407** may further package and/or otherwise transform the data to send via network stack **405** for delivery via network **506** to execution environment **401** of second node **504**.

[0135] Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in a communication with the communications agent in execution environment **401** of second node **504**.

[0136] Returning to FIG. **2A**, a block **206** illustrates that the method yet further includes receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. Accordingly, a system for processing a data object identification request in a communication includes means for

receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. For example, the arrangement in FIG. **3A**, includes result handler component **306** that is operable for receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. FIG. **4** illustrate result handler components **406** as an adaptations and/or analogs of result handler component **306** in FIG. **3A**. One or more result handler components **406** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. In FIG. **4**, a result handler component **406** is illustrated as a component of application **403**.

[0137] FIG. **7** includes dataflow message **714** that illustrates a second message, including a data object identification response, received via network **506** by execution environment **401** of first node **502** from execution environment **401** of second node **504**.

[0138] In addition to packaging communication information including a data object identification request into a valid communication, content manager component **409** operating in execution environment **401** of first node **502** may interoperate with communications protocol layer component **407** and/or network stack **405** to receive data in one or more communications via network **506** with another communications agent in another node. A result handler component **406** may receive some or all of the

communication in receiving and/or otherwise identifying a data object identification response in the communication. Content manager component **409** in FIG. **4** operatively couples result handler component **406** with network **506**, via com-in component **411**, communications protocol component **407**, and/or network stack **405** to receive the data from the other node. Thus, execution environment **401** of first node **502** may receive a communication, via content manager component **409** and result handler component **406**, that includes a data object identification response that is a response to a data object identification request detected by execution environment **401** of second node **504** in a message received previously from the execution environment **401** of first node **502**.

[0139] Com-in component **411** may provide data received in a communication to content manager component **409**. Content handler component **409** may identify a data object identification response in the communication sent from the execution environment **401** of second node **504**. Content manager component **409** may interoperate with result handler component **406** in identifying the data object identification response. The data object identification response may be identified by content manager component **409** according to a schema for the communication. For example, a portion of an email message may include a MIME-type identifier that identifies a corresponding part of the email as a data object identification response and/or otherwise identifies the part for routing to a result handler component **406** in communications agent **403** in execution environment **401** of first node **502** for detecting the data object identification response.

[0140] Data sent in second message **714**, by execution environment **401** of second node **504**, may be received by com-in component **411** operating in an instance, adaptation, and/or analog of execution environment **401** including and/or provided by first node **502**. The received communication includes a message addressed to the a communicant represented by communications agent **403** in first node **502** and a data object identification response identifying a data object located and/or otherwise identified based on the data object identification request identified by first message **706**. Content manager component **409** may detect the user message, such as message portion **802b** in second message **714**. Content manager component **409** may detect a data object identification response, such as data object identification response portion

804b in second message **714**. Message portion **802b** and data object identification response portion **804b** may be provided to suitable content handler components **413** based on the content types of the message portion and the data object identification response portion detected by content manager component **409**. In FIG. **4**, data object identification response portion **804b** may be provided to result handler component **406** as a content handler that matches the MIME type field **806b** illustrated in FIG. **8A**.

[0141] In an aspect, a data object identification response that is a response to a data object identification request sent in a previous message may include information identifying one or more data objects located and/or otherwise identified by the execution environment **401** of second node **504** that processed the data object identification request. For example, a data object identification response may be sent that identifies a URI for each data object located and/or otherwise identified by a first node in response to processing a corresponding data object identification request.

[0142] Returning to FIG. **2A**, a block **208** illustrates that the method yet further includes sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. Accordingly, a system for processing a data object identification request in a communication includes means for sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. For example, the arrangement in FIG. **3A**, includes access request constructor component **313** that is operable for sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. FIG. **4** illustrates access request constructor component **408** as an adaptation and/or analog of access request constructor component **308** in FIG. **3A**. One or more access request constructor components **408** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. In FIG. **4**, a access request

constructor component **408** is illustrated as a component of result handler component **406**.

[0143] FIG. **7** includes dataflow message **716** that illustrates a present message that may be internal to execution environment **401** of first node **502** and/or may be a message sent to a remote device, such as a node including a browser, to send presentation information to present, via an output device, a representation of the data object identification response received in second message **714**. FIG. **7** also includes dataflow message **718** that illustrates a create access request message that may be a data flow internal to execution environment **401** of first node **502** and/or a message received via network **506** by execution environment **401** of first node **502**. In the dataflow message **718** an ARC component **408** operating in execution environment **401** of first node **502** may be invoked to operate based on result information. Result information may be identified to and/or otherwise by ARC component **408** as part of and/or in response to being invoked.

[0144] Result handler component **406** may be a type of content handler component **413** that operates to process data representations and may provide transformed data from the representations to one or more user interface element handler components **415**. Result content handler component **406** may operate to process data object identification responses according to a schema defining valid data object identification responses. Content manager component **409** may provide and/or otherwise identify various portions of a communication, such as a data object identification response, to one or more content handler components **413**. The data object identification response may be provided to result handler component **404**. In an aspect, result handler component may interoperate with the presentation controller component **417** in the execution environment **401** of first node **502** to present the data received in the data object identification response and/or other data received in the communication.

[0145] FIG. **6B** illustrates a response window UI element **602b** as an exemplary user interface that may be presented by communications agent **403** operating in execution environment **401** of first node **502** to present data received in a communication from execution environment **401** of second node **504**. FIG. **6B** illustrates a contactor UI element **604b**, a contactee UI element **606b**, a presentation space **608b** of the

response window UI element **602b**. Presentation space **608** includes a user message UI element **610b**, analogous to similar UI elements described with respect to FIG. **6A**. The various UI elements may be presented by one or more UI element handler components **415** in the communications agent **402** in execution environment **401** of first node **502**. A reply UI element **622b** and a reply all UI element **624b** illustrate UI elements in presentation space **608b** presented to allow a user of the execution environment **401** of first node **502** to respond and/or otherwise instruct execution environment **401** of first node **502** to perform an operation based on the message received in the communication. FIG. **6B** illustrates a data object identification requested query UI interface element **626b** where the presented term "None" may be defined to indicate no data object identification request was detected by the content manager component **409** in execution environment **401** of first node **502** in the communication received by execution environment **401** of first node **502** from second node **504** via network **506**.

[0146] Response window UI element **602b** in FIG. **6B** includes a data object identification response UI element **628b** for receiving user input to display a representation of one or more data objects located and/or otherwise identified by the execution environment **401** of second node **504** and identified in the data object identification response. One or more of the data objects is not included in the message received in the communication. In response to an input directed to data object identification response UI element **628b**, a corresponding UI element handler **415** may interoperate with result handler component **406** to present a located data resources dialog UI element **630b** that includes a presentation space **632b** for presenting a user detectable representation of some or all of the data objects identified in the received data object identification response from the execution environment **401** of second node **504**. In an aspect, one or more data objects located and/or otherwise identified by execution environment **401** of second node **504** may be included in a communication as attachments. In a further aspect, located and/or otherwise identified data objects not included in the message may be identified as described with respect to located and/or otherwise identified data objects dialog UI element **630b** and/or in any other suitable manner. In yet another aspect, files and/or other data entities identified in the data

object identification response may be illustrated in a manner similar to attachments rather than in a separate UI element as in FIG. **6B**, and accessed from first node **504** via network **506** as described below.

[0147] FIG. **6B** illustrates an access UI element **634b** and an access all UI element **636b** allowing a user to provide user input(s) to instruct communications agent **403** in execution environment **401** of first node **502** to request one or more data objects identified in the data object identification response.

[0148] FIG. **7** includes a dataflow message **720** that illustrates a third message that includes an access request to access a data object identified in the data object identification response received in second message **714**. FIG. **7** illustrates the access request is sent via network **506** by execution environment **401** of first node **502** to execution environment **401** of second node **504**.

[0149] In response to receiving second message **714**, execution environment **401** of first node **502** may present, in a manner described above, data received in the message to a user of execution environment **401** of first node **502** as described with respect to FIG. **6C**. Context menu **646c** provides UI elements to allow the user to instruct execution environment **401** of first node **502** to retrieve a data object identified in a data object identification response received from execution environment **401** of second node **504**. In response to a user input corresponding to a context menu item included in context menu **646c** a corresponding UI element handler component **415** may invoke ARC component **408** to construct an access request to retrieve one or more data objects identified in the data object identification response. FIG. **6G** illustrates another exemplary user interface that execution environment **401** of first node **502** may present to the user to allow the user to instruct execution environment **401** of first node **502** to construct and send a request. Auto-response window **602g** may be presented in response to a data object identification response received in a communication without a user message. Rather than a UI element for displaying a user message, a results pane UI element **670g** may be presented that identifies data objects located and/or otherwise identified by execution environment **401** of second node **504** that may be retrieved by execution environment **401** of first node **502**. In an aspect, execution environment **401** of second node **504** may process a data object identification request and automatically

send a data object identification response without involving the user of execution environment **401** of second node **504**. The data object identification response may be received without a user message as FIG. **6G** illustrates or may be received with a pre-written and/or automatically generated message. UI elements **672g-674g** may be presented by execution environment **401** of first node **502** to allow the user of execution environment **401** of first node **502** to request the retrieval of one or more data objects identified in the data object identification response.

[0150] ARC component **408** may be invoked to construct an access request to retrieve one or more data objects identified in the data object identification response. An access request may be based on a URL, such as an HTTP and/or FTP URL that identifies and locates a data object. Alternatively or additionally, an access request may be sent in a communication supported by a communications agent **403**, in a manner analogous to that illustrated with respect to FIG. 9A. See U. S. Patent Application No. 12/833,014 (published US 2012-0011207 A1) filed on 2010/07/09, entitled "Methods, Systems, and Program Products for Processing a Request for a Resource in a Communication" and Application No. 12/833,016 (published US 2012-0011444 A1) filed on 2010/07/09, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR REFERENCING AN ATTACHMENT IN A COMMUNICATION". As described above, content manager component **409** in execution environment **401** of first node **502** may package the one or more data representations including a representation of an access request into a message in a communication. The message is formatted according to a communications protocol supported by communications agent **403** in execution environment **401** of first node **502**. Communications protocol component **407** operating in execution environment **401** of first node **502** may send the message according to the specification(s) of the communications protocol. Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in a communication.

[0151] Content manager component **409** operating in execution environment **401** included in and/or otherwise provided by execution environment **401** of first node **502** may provide the packaged, encoded, and/or transformed data to communications

protocol component **407** via a com-out component **419**. Com-out component **419** as described above operatively couples communications agent **409** to communications protocol component **407** according to an interface provided by communications protocol component **407** for sending data in a communication according to a communications protocol. Communications protocol component **407** may further package and/or otherwise transform the data to send via network stack **405** for delivery via network **506** to execution environment **401** of second node **504**.

[0152] Returning to FIG. **2A**, a block **210** illustrates that the method yet further includes receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. Accordingly, a system for processing a data object identification request in a communication includes means for receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. For example, the arrangement in FIG. 3, includes content director component **310** that is operable for receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. FIG. **4** illustrates content director component **410** as an adaptation and/or analog of content director component **310** in FIG. 3. One or more content director components **410** operate in an execution environment **401**. The system for content director includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. In FIG. **4**, a content director component **410** is illustrated as a component of communications agent **403**.

[0153] FIG. **7** includes dataflow message **722** that illustrates a fourth message that includes an access response received via network **506** by execution environment **401** of first node **502** from execution environment **401** of second node **504**.

[0154] As described above, content manager component **409** operating in execution environment **401** of first node **502** may interoperate with communications protocol layer component **407** and/or network stack **405** to receive data in a communication, via network **506**, with another communications agent in another node. Content manager

component **409** is operatively coupled, via com-in component **411**, to communications protocol component **407** to receive the data from the other node. Thus, execution environment **401** of first node **502** may receive a message in a communication with first node **504** via a com-in component **411**. The message includes an access response to an access request. Content director component **410** may be included in content manager component **409** to detect one or more portions of a received message, such as a portion that is a response to an access request. Content director component **410** may invoke one or more content handler components **413** to process one or more data objects provided in the message according to their type.

[0155] In one aspect, a data object returned in a message in response to access request may be processed as an attachment and/or in a manner analogous to processing of an attachment for a particular communications type.

[0156] With reference to FIG. **2B**, a block **212** illustrates that the method includes receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. Accordingly, a system for processing a data object identification request in a communication includes means for receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. For example, the arrangement in FIG. **3B**, includes content manager component **309** that is operable for receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. FIG. **4** illustrates content manager component **409** as an adaptation and/or analog of content manager component **309** in FIG. **3B**. One or more content manager components **409** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, via a network by a second communications agent

representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. In FIG. **4**, a content manager component **409** is illustrated as a component of communications agent **403**.

[0157] As described above, FIG. **7** illustrates first message **706**, including a data object identification request, received via network **506** by execution environment **401** of second node **504** from execution environment **401** of first node **502.**

[0158] In addition to packaging representations of data in a structure and/or format for sending a message according to a communications protocol, a content manager component **409** may operate to detect a data object identification request received and/or otherwise identified in a message received in a communication. Content manager component **409** operating in execution environment **401** of second node **504** may receive the message, illustrated by first message **706**, in a communication with execution environment **401** of first node **502**. Com-in component **411** in execution environment **401** of second node **504** may receive the message via communication protocol component **407** and network stack **405**. First message **706** may be delivered to execution environment **401** of second node **504** via network **506** based on a communications address of a communicant represented by communications agent **403** in execution environment **401** of second node **504**. A communications agent may represent more than one communicant, which may be identified by different addresses.

[0159] The first message **706** in FIG. **7** may be received in one or more packets via network **506** by network stack **405** and communications protocol component **407** in an instance of and/or analog of execution environment **401** including and/or otherwise provided by execution environment **401** of second node **504**. The data in the communication may be received by com-in component **411**. Com-in component **411** may provide the data to content manager component **409**. Content manager component **409** may determine one or more content types of the data. The content and/or portions of the content may be provided to one or more content type handler components **413** based on the one or more content types identified by content manager component **409**. For example, FIG. **8A** illustrates message portion **812a** including “text/plain” MIME type identifier **814a** as a content type identifier. Message portion **812a** may be provided to a

text/plain content handler **413**. Audio data in a voice communication may be provided to an audio content handler component **413**, and video data in a video communication may be provided to a video content handler component **413**.

[0160] Returning to FIG. **2B**, a block **214** illustrates that the method further includes detecting, by the second communications agent, a data object identification request included in the first message. Accordingly, a system for processing a data object identification request in a communication includes means for detecting, by the second communications agent, a data object identification request included in the first message. For example, the arrangement in FIG. **3B**, includes query handler component **302** that is operable for detecting, by the second communications agent, a data object identification request included in the first message. FIG. **4** illustrates query handler component **402** as an adaptation and/or analog of query handler component **302** in FIG. **3B**. One or more query handler components **402** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for detecting, by the second communications agent, a data object identification request included in the first message. In FIG. **4**, query handler component **402** is illustrated as a component of communications agent **403**.

[0161] FIG. **7** includes a dataflow message **708** that illustrates a detect data object identification request dataflow that may be internal to execution environment **401** of second node **504** and/or may include interoperation with another node via a network. With respect to FIG. **4**, dataflow message **708** may correspond to a data exchange and/or other interoperation between content handler component **409** and query handler component **402** to detect and/or otherwise process the data object identification request received in first message **706** by execution environment **401** of second node **504**. FIG. **7** further includes a dataflow message **710** that illustrates a search message for locating one or more data objects based on the data object identification request. The data object identification request may include and/or otherwise identify a data object matching criterion identified by data object information identified by the communicant represented by execution environment **401** of first node **502**. The one or more data

objects may be located and/or otherwise identified by determining a data object or data objects that match the data object matching criterion. The data object(s) may be in a data store accessible to execution environment **401** of second node **504**, such as one or more files in a file system of execution environment **401** of second node **504**. Dataflow message **710** may be internal to execution environment **401** of second node **504** and/or may be a message sent to a remote device, such as database server and/or a network attached storage device providing storage locations for some or all of a file system of execution environment **401** of second node **504**.

[0162] Com-in component **411** in execution environment **401** of second node **504**, as described above, provides the message, received from execution environment **401** of first node **502**, to content manager component **409**. Content manager component **409** is operatively coupled, via com-in component **411**, to communications protocol component **407** for receiving data in communications with other nodes, such as execution environment **401** of first node **502**. Thus, execution environment **401** of second node **504** may receive a message via a com-in component **411** in a communication with execution environment **401** of first node **502**. The message may include a data object identification request based on data object information identified by a user to communications agent **403** in execution environment **401** of first node **502**.

[0163] Data received in a communication, which may for example include a voice session and/or an instant message, may include data of various content types. In FIG. **4**, communications agent **403** in execution environment **401** of second node **504** includes one or more content handler components **413** to process data received according to its content type. Content manager component **409** may detect and/or otherwise identify a portion of the message as the data object identification request sent from execution environment **401** of first node **502**. The data object identification request may be identified by content manager component **409** according to a schema for the message and/or a schema for the data object identification request. For example, a portion of the message may include a MIME-type identifier that identifies the portion as a data object identification request and/or otherwise identifies the portion for routing to a query handler component **402** operating in and/or otherwise on behalf of execution environment **401** of second node **504**. Alternatively or additionally, the data object

identification request may have structure and/or content that is valid and identifiable according to a schema for defining and/or otherwise identifying valid data object identification requests. A query handler component, may be provided as a type of content handler component, in an aspect. Alternatively or additionally, a query handler component may interoperate with one or more content handler components to detect a data object identification request in a message received by a content manager component.

[0164] As such, query handler component **402** may operate along with other content handler components **413** to process data representations received in the message **706** from execution environment **401** of first node **502**. Query handler component **402** may also provide transformed data from the representations to one or more user interface element handler components **415**. Query handler component **402** operates to validate and process data object identification requests according to a schema defining valid data object identification requests. Various portions of a message including a data object identification request may be provided to one or more content handler components **413**, including query handler component **402**, to interoperate with presentation controller component **417** in execution environment **401** of second node **504** to present some or all of the received message, including the data object identification request.

[0165] FIG. **6D** illustrates a window, request window UI element **602d**, presenting an exemplary representation of a message received from execution environment **401** of first node **502** in a communication with second node **504**. As in FIG. **6A**, request window UI element **602d** includes a contactor UI element **604d**, a contactee UI element **606d**, a presentation space **608d** of the request window UI element **602d** that includes a user message UI element **610d** including a text portion of the message provided by the communicant represented by execution environment **401** of first node **502**. The various UI elements may be presented by one or more UI element handler components **415** in communications agent **403** operating in execution environment **401** of second node **504**. A reply UI element **648d** and a reply all UI element **648d** are illustrated as exemplary UI elements in presentation space **608d** presented to allow a user of execution environment **401** of second node **504** to respond and/or otherwise perform an

operation based on the received message. FIG. **6D** illustrates a data object identification request UI interface element **652d** where the data object identification request identified in the message from execution environment **401** of first node **502** is represented to the communicant represented by the execution environment **401** of second node **504**. In an aspect, the communicant is allowed to authorize the request by providing an input corresponding to a search and reply UI element **654d**. In another aspect, the communicant may be allowed to modify the received data object identification request.

[0166] FIG. **6E**, illustrates a reply window UI element **602e** that, in an aspect, may be presented by communications agent **403** in execution environment **401** of second node **504** in response to an user input corresponding to reply UI element **648d** detected by a corresponding UI element handler component **415**. In an aspect, a user of communications agent **403** operating in in execution environment **401** of second node **504** may provide input authorizing some or all of the processing of a data object identification request. Reply window UI element **602e** in FIG. **6E** includes a search UI element **656e** for receiving user input authorizing and/or otherwise instructing execution environment **401** of second node **504** to locate data objects that match the data object identification request, illustrated in a data object identification requested UI element **658e**.

[0167] As described herein in an aspect, content manager component **409** may detect content type information to detect a data object identification request in a communication. For example, the message portion illustrated in FIG. **8A** may be received in first message **706**. Content manager component **409** may detect "application/id-request" MIME type identifier in content-type header **804a**. The "application/id-request" MIME type identifier may be defined to identify a data object identification request based on one or more matching criteria for identifying a requested data object. Content manager component **409** may identify data object identification request portion **802a** as including the data object identification request.

[0168] In response to detecting a data object identification request in first message **706**, content manager component **409** may provide some or all of the data object identification request to query handler component **402**. For example, query handler

component **402** may be configured to operate according to a schema defining a format and/or a vocabulary for an XML-based language for id-request documents. Content manager component **409** may provide id-request document **806a**, as a data object identification request, to query handler component **402**. Query handler component **402** may operate according to the id-request schema. In an aspect, a query handler component may process more than one data object identification request content type. Alternatively or additionally, execution environment **401** may include multiple query handler components **402** for supporting multiple data object identification request content types.

[0169] Returning to FIG. **2B**, a block **216** illustrates that the method yet further includes generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. Accordingly, a system for processing a data object identification request in a communication includes means for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. For example, the arrangement in FIG. **3B**, includes query director component **312** that is operable for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. FIG. **4** illustrates query director component 412 as an adaptation and/or analog of query director component **312** in FIG. **3B**. One or more query director components **412** operate in an execution environment **401**. The system for query director includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is

identified by processing the data object identification request. In FIG. **4**, a query director component **412** is illustrated as a component of application **403**.

[0170] FIG. **7** includes dataflow message **712** that illustrates a create data object identification response data flow to process results returned from search dataflow **710**. In FIG. **4**, dataflow message **712** in FIG. **7** may represent data exchanged between query handler component **402** and query director component **412** to process results returned from a search operation performed based on the data object identification request, as illustrated by dataflow message **710**. Processing the results may include creating and/or otherwise constructing a data object identification response to the data object identification request identified in first message **706**. Dataflow message **712** may be internal to execution environment **401** of second node **504** and/or may include interoperation by execution environment **401** of second node **504** via a network with another node.

[0171] FIG. **6F**, illustrates an identify in response dialog UI element **660f** that, in an aspect, may be presented by communications agent **403** in execution environment **401** of second node **504** in response to processing the data object identification request to locate matching data objects. In an aspect, a user of communications agent **403** operating in execution environment **401** of second node **504** may provide input corresponding to search UI element **656e** in reply window UI element **602e** in FIG. **6E**. A corresponding UI element handler component **415** may operate to invoke query director component **412** to perform and/or otherwise to provide for performing one or more operations to locate one or more data objects based on the data object identification request. Query director component **412** may interoperate with one or more UI element handlers **415**, directly or indirectly via query handler component **402**, in an aspect, to present the UI element **660f** and its contents. Identify in response dialog UI element **660f** includes results pane UI element **662f** including representations of two data objects located and/or otherwise identified by query director component **412** based on the data object identification request. A user may select one or more data objects to identify in a data object identification response. An include UI element **664f** may be presented, as illustrated, to allow a user to instruct communications agent **403** to identify one or more selected data objects in results pane UI element **662f** in a data

object identification response. An include all UI element **666f** is illustrated to allow for user input instructing communications agent **403** to identify all data objects represented in the UI element **660f** in a data object identification response in a communication with execution environment **401** of first node **502**. A cancel UI element **668f** allows a user to end presentation of dialog UI element **660f**. A user may opt to identify no data objects represented in the dialog UI element **660f** and located and/or otherwise identified by query director component **412** based on the data object identification request.

[0172] Data objects located and/or otherwise identified by query director component **412** may be identified to IRB component **414** to construct and/or otherwise create a data object identification response identifying the data objects. As described above, in an aspect, a user may be allowed to select which data objects, if any, are to be identified in a data object identification response from one or more data objects located and/or otherwise identified by query director component **412** based on the data object identification response. In another aspect, communications agent **403** in FIG. **4** may invoke query director component **412** automatically to locate one or more data objects based on a data object matching criterion identified in a data object identification request and in response to detecting a data object identification request in a communication. In another aspect, communications agent **403** may invoke query director component **412** automatically to locate one or more data objects based on a data object identification request in a communication in response to detecting an indication to present, via a output device, some or all of a user message received in a message in the communication. Query director component **412** may be invoked and/or may operate based on the data object identification request and/or data object matching criterion identified to locate a data object prior to, during, and/or after other content received in the communication is processed.

[0173] The method illustrated in FIG. **2B** may include additional aspects supported by various adaptations and/or analogs of the arrangement of components in FIG. **4**. As described above, in one aspect a data object may be located and/or otherwise identified automatically in response to detecting the data object identification request. In another aspect, locating a data object may include presenting a locator UI element representing a data object identification request. Search UI element **656e** in FIG. **6E** exemplifies one

type of locator UI control. Search UI element **656e** may be presented by presentation controller component **417** as described above. A user input corresponding to a search UI element, such as search UI element **656e**, may be detected by presentation controller component **417**. Presentation controller component **417** may instruct query director component **412** to locate the data object based on a data object identification request represented by the data object identification request UI element **658e**.

[0174] In another aspect, a data object identification request UI element may be presented to receive one or more additional matching criteria from a user. Based on the data object identification request UI element and input from the user, query director component **412** may receive one or more additional data object matching criterion information and/or may receive input to modify a data object matching criterion identified in a received data object identification request. Query director component **412** may locate one or more data objects in response to receiving the additional criterion.

[0175] Returning to FIG. **2B**, a block **218** illustrates that the method yet further includes sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. Accordingly, a system for processing a data object identification request in a communication includes means for sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. For example, the arrangement in FIG. **3B**, includes identification response builder component **314** that is operable for sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. FIG. **4** illustrates identification response builder component **414** as an adaptation and/or analog of identification response builder component **314** in FIG. **3B**. One or more identification response builder components **414** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for sending a

second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. In FIG. **4**, a identification response builder component **414** is illustrated as a component of application **403**.

[0176] As described above, FIG. **7** includes second message dataflow message **714** illustrating second message sent via network **506** to execution environment **401** of first node **502** from execution environment **401** of second node **504**. Second message **714** includes a data object identification response in response to the data object identification request from first node **502** received and/or otherwise identified in first message **706**.

[0177] IRB component **414** may provide the data object identification response to content manager component **409** to include in a message, illustrated by second message **714**. As described above, content manager component **409** may receive and/or otherwise identify various portions of a message, including a data object identification response, from one or more content handler components **413**. The data object identification response may be provided to content manager component **409**. As described above, content manager component **409** in execution environment **401** of second node **504** may package the one or more data representations including a representation of the data object identification response into the message formatted according to a suitable communications protocol. Communications protocol component **407** operating in execution environment **401** of second node **504** may send the data as a message in a communication with execution environment **401** of first node **502** according to the specification(s) of the communications protocol. Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in the communication to the communications agent.

[0178] FIG. **8B** illustrates an exemplary data object identification response described in detail above. Second message portion **802b** illustrates a user message portion that may be included in second message **714**. Note that a user message portion may be empty or absent in some aspects. Data object identification response portion **804b** illustrates

the data object identification response as a list of URIs identifying data objects in a file system of execution environment **401** of second node **504**. User Message portion **802b** and data object identification response portion **804b** may be provided by respective content handler components **413** to content manager component **409**. Content manager component **409** may construct the content as illustrated in FIG. **8B** and/or otherwise provide for encoding, translating, combining, and/or otherwise preparing the content for sending in second message **714**.

[0179] The methods illustrated in FIG. **2A-B** may include additional aspects supported by various adaptations and/or analogs of the arrangement of components in FIG. **3A-B**. In various aspects, a data object identification request may be based on a date, a time, a length of time, a file type, a database record key, content of the data object, a content type identifier, a format rule, a vocabulary, a role of a user, a security attribute, a location in a data store such as a file system, an attribute of an identified data object, a size, a task, a transaction, a state, a user, a group, a requester, a relationship including a requesting user and a responding user, a keyword, a tag, a folder, and/or a path portion of a data object identifier – to name a few examples.

[0180] In another aspect, a data object located and/or otherwise identified in response to receiving a data object identification request must meet match a query identified by the data object identification request. Alternatively or additionally, a data object identification request may include an instruction and/or input for generating a data object. Locating the data object may include generating the data object. Generating a data object may include creating the data object and/or may include modifying and/or otherwise transforming an existing data object. For example, execution environment **401** of second node **504** may include a template stored in a file system. A data object identification request may identify the template. A query handler component **402** may operate to identify a document or other data object that may be created, is being created, and/or has been created based on the template. In response to an access request, execution environment **401** of second node **504** may return the document.

[0181] As described above, in various aspects, a data object identification request may be detected and/or represented based on various syntaxes, grammars, vocabularies, and/or languages. For example, a data object identification request may be identified

and/or represented according to a file system search syntax, a regular expression language, a structured query language (SQL) query, a universal data object identifier schema, an XPATH based language, an XQuery based language, an XML based language, an HTML based language (form-based), and/or a keyword-value pair based language.

[0182] A data object identification request in a communication may be communicated via a network according to a first communications protocol. A user message may be exchanged between communicants in the communication via a second communications protocol. For example, a data object identification request may be sent on behalf of a first communicant by execution environment **401** of first node **502** including a communications agent **403** representing the first communicant. The data object identification request may be included in a communication along with a text message sent to the execution environment **401** of second node **504**. The communicants represented by the execution environment **401** of first node **502** and the execution environment **401** of second node **504** may communicate via one or more exchanges of audio exchanged via, for example, a voice over IP (VoIP) communications protocol.

[0183] Exemplary data objects that may be requested via a data object identification request include a file, a program component, a data base record, video data, audio data, markup language, binary data, text data, an output of a service. Requested data objects may be pre-existing, volatile, and/or generated in response to the request.

[0184] As described above, receiving data object information may include receiving, via network from a node that received the data object information from the user, a message that identifies the data object information. For example, data object information may be received from a browser by a web server.

[0185] Data object information and/or a data object identification request may be generated, detected, and/or otherwise processed according to a schema that identifies at least one of a rule and a vocabulary that defines a valid data object identification request. Similarly, a data object identification response, an access request, and a response to an access request may be generated, detected, and/or otherwise processed according to a schema. One or more schemas may alone or together define one or more of valid data object information, a data object identification request, data

object identification response, an access request, and/or an access response. A schema, as just described, may be identified based on data received and/or otherwise identified by a node hosting a communications agent via the network from another node hosting a communications agent and/or from a path node included in a network path in the network communicatively coupling the node to one or more nodes that respectively host communications agents.

[0186] In a further aspect, multiple alternative schemas may be defined for each of data object information, data object identification requests, data object identification responses, access requests, an/or an access responses processing data object identification requests. A schema may be selected and/or otherwise identified from the multiple schemas by execution environment **401** of first node **502** and the execution environment **401** of second node **504** and/or based on information exchanged by the two nodes. For example, a schema may be selected by the first communicant and/or the second communicant. One may inform the other, via a communications protocol and/or via any other suitable means. The communicants may negotiate which schema to use via any suitable means.

[0187] The communications agents in execution environment **401** of second node **504** and execution environment **401** of first node **502** may select a schema For example, a schema may be identified and/or otherwise selected based on a communications protocol supported and/or a communications protocol not supported by one or both nodes. That is, in negotiating which communication protocol to use allowing communicants to exchange information, a schema may be identified. A schema may be identified and/or otherwise selected based on an attribute of the data object. A schema for identifying image data may be different than a schema for editable documents. A schema for identifying a data object in a first file system may be different than a schema for identifying a data object in a different file system.

[0188] As described above, data object information and/or a data object identification request may identify a scope that specifies all or some portion of one or more data stores for locating a data object. For example, A communications agent **403** in execution environment **401** of first node **502** may receive user input that identifies a

folder in a file system of the execution environment **401** of second node **504** that defines a scope for locating a data object.

[0189] Data object information and/or a data object identification request may be represented in a representation that includes a complete or a portion of at least one of a keyword expression, a regular expression, an XQuery expression, an XPath expression, a file system path expression, and a structured query language statement.

[0190] Further as also described above, a data object identification request, a data object identification response, an access request, an/or an access response may be included in a message according to a schema that identifies the message as valid for a specified communications protocol. Those skilled in the art are aware that schemas for email differ from schemas for instant messages and from streamed audio, for example.

[0191] A data object identification request, a data object identification response, an access request, and/or an access response may be exchanged in a communication in which audio data and/or image data is also exchanged. Multimedia may be a packaged to include one or more data streams which may include text based data streams.

[0192] A data object identification request, a data object identification response, an access request, and/or an access response may be included in an email, an instant message, a multi-media message, a short-message service message, and/or a data stream. The data stream may include audio data, image data, and/or text data.

[0193] A data object identification request, a data object identification response, an access request, an/or an access response may be included in and/or detected in a communication based on a location in the communication and/or based on a marker, such as an identifier in the communication that identifies a location for the data object identification request, the data object identification response, the access request, and/or the access response. The location may be absolute, such as fixed number of bytes or fields from the start or end of a message. Alternatively or additionally, the location may be identified by a location relative to another detectable portion of the communication. FIGs. **8A-B** illustrate the use of MIME type identifiers as markers that identify locations for one or more of a data object identification request, a data object identification response, an access request, an/or an access response in a communication.

[0194] A data object identification request, a data object identification response, an access request, an/or an access response or respective portions thereof may include at least a portion of a keyword expression, a regular expression, expression including a Boolean operator, an expression including a precedence information, and a structured query language statement.

[0195] With respect to the method illustrated in FIG. **2B**, the method as described above may include locating and/or otherwise identifying a data object automatically in response to receiving an access request that matches and/or otherwise identifies the data object. Further, a data object identification response may be created and/or otherwise generated automatically, in response to locating and/or otherwise identifying the data object.

[0196] In an aspect, in response to receiving and/or otherwise detecting a data object identification request, a user interface may be presented based on the data object identification request to prompt a user to process the data object identification request in some manner, such as authorizing the request and/or modifying the request by restricting its scope. In response to detecting user input targeting and/or otherwise corresponding to the presented user interface, one or more data objects may be located and/or otherwise identified based on the data object identification request.

[0197] A user may change a received data object identification request. One or more data objects may be located and/or otherwise identified based on the changed data object identification request.

[0198] In a further aspect, once one or more data objects are located and/or otherwise identified based on a data object identification request, a user interface may be presented to allow a user to identify the one or more data objects. User information may be received that selects one or more data objects to identify in a data object identification response and/or to not identify in the data object identification response. A user, in an aspect, may be allowed to identify a data object not located and/or otherwise identified by the user's node. The data object, based on the user input, is identified in the data object identification response.

[0199] In light of the description provided above, a data object identification response may be generated automatically in response to detecting a corresponding data object

identification request by an execution environment **401** of first node **502**. Further, the automatically generated data object identification response, may be sent to the node that sent the corresponding data object identification request, automatically in response to the generating of the data object identification response.

[0200] With respect to the method illustrated in FIG. **2A**, the third message may be sent automatically in response to detecting the data object identification response. In another aspect, a representation of a received data object identification response may be presented, via an output device, to a user. The representation may identify one or more data objects located and/or otherwise identified based on the data object identification request. The representation may be presented automatically, in response to detecting the data object identification response. Further, a user input may be detected that corresponds to and/or otherwise targets the representation. The user input may be processed as an indication to retrieve the data object. As a result an access request may be created or otherwise generated that identifies the data object. The access request may be sent in a message via the network to retrieve the data object. The message may be sent to a node that sent the data object identification response and/or to another node. The third message may be sent by the communications agent in the execution environment **401** of first node **502** to the communications agent in the execution environment **401** of second node **504** via a communications protocol. The third message sent via the communication protocol may include a communications address, in an address space of the communications protocol, that identifies a user represented by the communications agent in the execution environment **401** of second node **504**. The third message may include a data object identification request, a data object identification response, and/or a data object sent in response to previously receiving an access response by the execution environment **401** of first node **502**. The third message may include an access request that identifies the data object. The access request may include a universal data object identifier (URI) that identifies the data object.

[0201] With respect to the method illustrated in FIG. **2B**, in response to receiving an access request, a data object identified in the request may be sent. The data object may be sent automatically, in response to receiving the access request. The data object may

be sent as attachment in a communication. The data object may be sent by a communications agent in the execution environment **401** of second node **504** to a communications agent in the execution environment **401** of first node **502** via a communications protocol. The data object may be sent via a communication that includes a communications address, in an address space of the communications protocol, that identifies the first communicant represented by the execution environment **401** of first node **502**. The communication that includes the data object may include a data object identification request, a data object identification response, and/or an access request sent in response to a data object identification response sent previously by the first node.

[0202] The methods illustrated in FIGs. **2A-B** may yet further include additional aspects supported by various adaptations and/or analogs of the arrangement of components in FIGs. **3A-B**, respectively.

[0203] Performing the method illustrated in FIG. **2A** and/or the method illustrated in FIG. **2B** and/or any of its extension and/or in any of its aspects may include one or more of calling a function or method of an object, sending a message via a network; sending a message via an inter-process communication mechanism such as a pipe, a semaphore, a shared data area, and/or a queue; and/or receiving a request such as poll and responding to invoke, and sending an asynchronous message.

[0204] Output devices suitable for presenting a representation of a data object and/or a tag include a visual output device, an audio output device, and a tactile output device. One output device may present a data object and another output device may present a tag with which the data object is tagged.

[0205] Any action included in performing the method illustrated in FIG. **2A** as well as any action included in in performing the method illustrated in FIG. **2B** may include sending and/or receiving a message via a network. Further, a message, included in performing any of the subject matter described herein and/or any of its extensions in any of its aspects, may an asynchronous message without a corresponding request.

[0206] To the accomplishment of the foregoing and related ends, the descriptions and annexed drawings set forth certain illustrative aspects and implementations of the disclosure. These are indicative of but a few of the various ways in which one or more

aspects of the disclosure may be employed. The other aspects, advantages, and novel features of the disclosure will become apparent from the detailed description included herein when considered in conjunction with the annexed drawings.

[0207] It should be understood that the various components illustrated in the various block diagrams represent logical components that are configured to perform the functionality described herein and may be implemented in software, hardware, or a combination of the two. Moreover, some or all of these logical components may be combined, some may be omitted altogether, and additional components may be added while still achieving the functionality described herein. Thus, the subject matter described herein may be embodied in many different variations, and all such variations are contemplated to be within the scope of what is claimed.

[0208] To facilitate an understanding of the subject matter described above, many aspects are described in terms of sequences of actions that may be performed by elements of a computer system. For example, it will be recognized that the various actions may be performed by specialized circuits or circuitry (e.g., discrete logic gates interconnected to perform a specialized function), by program instructions being executed by one or more processors, or by a combination of both. The description herein of any sequence of actions is not intended to imply that the specific order described for performing that sequence must be followed.

[0209] Moreover, the methods described herein may be embodied in executable instructions stored in a non-transitory computer readable medium for use by or in connection with an instruction execution machine, system, apparatus, or device, such as a computer-based or processor-containing machine, system, apparatus, or device. As used here, a "non-transitory computer readable medium" may include one or more of any suitable media for storing the executable instructions of a computer program in one or more forms including an electronic, magnetic, optical, and electromagnetic form, such that the instruction execution machine, system, apparatus, or device may read (or fetch) the instructions from the non-transitory computer readable medium and execute the instructions for carrying out the described methods. A non-exhaustive list of conventional exemplary non-transitory computer readable media includes a portable computer diskette; a random access memory (RAM); a read only memory (ROM); an

erasable programmable read only memory (EPROM or Flash memory); optical storage devices, including a portable compact disc (CD), a portable digital video disc (DVD), a high definition DVD (HD-DVD.TM.), and a Blu-ray.TM. disc; and the like

[0210] Thus, the subject matter described herein may be embodied in many different forms, and all such forms are contemplated to be within the scope of what is claimed. It will be understood that various details may be changed without departing from the scope of the claimed subject matter. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation, as the scope of protection sought is defined by the claims as set forth hereinafter together with any equivalents.

[0211] Thus, the subject matter described herein may be embodied in many different forms, and all such forms are contemplated to be within the scope of what is claimed. It will be understood that various details may be changed without departing from the scope of the claimed subject matter. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation, as the scope of protection sought is defined by the claims as set forth hereinafter together with any equivalents.

[0212] All methods described herein may be performed in any order unless otherwise indicated herein explicitly or by context. The use of the terms "a" and "an" and "the" and similar referents in the context of the foregoing description and in the context of the following claims are to be construed to include the singular and the plural, unless otherwise indicated herein explicitly or clearly contradicted by context. The foregoing description is not to be interpreted as indicating that any non-claimed element is essential to the practice of the subject matter as claimed.

I CLAIM:

1. A computer-implemented method, comprising:

displaying, at a first node, at least one first interface with a first user interface element and a second user interface element, utilizing first hypertext markup language-equipped code that is received via at least one network;

receiving, utilizing the at least one first interface at the first node, a file description via the first user interface element that includes a textbox;

sending, from the first node via the at least one network, the file description;

receiving, utilizing the at least one first interface at the first node, an indication of at least one of a date or a time via the second user interface element;

sending, from the first node via the at least one network, the indication of the at least one of the date or the time;

displaying, at the first node, at least one second interface with a third user interface element and a fourth user interface element, utilizing second hypertext markup language-equipped code that is received via the at least one network;

receiving, utilizing the at least one second interface at the first node, indicia associated with at least one email address via the third user interface element;

sending, from the first node via the at least one network, the indicia associated with the at least one email address;

detecting, at the first node, a selection of the fourth user interface element;

sending, from the first node via the at least one network, an indication of the selection of the fourth user interface element;

after sending the indication of the selection of the fourth user interface element and based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; receiving, at a second node via the at least one network, a first message indicating the at least one of the date or the time, and including a fifth user interface element and at least one sentence including the file description;

detecting, at the second node, a selection of the fifth user interface element of the first message;

sending, from the second node via the at least one network, an indication of the selection of the fifth user interface element of the first message;

after sending the indication of the selection of the fifth user interface element, displaying, at the second node, at least one third interface with a sixth user interface element and a seventh user interface element, utilizing third hypertext markup language-equipped code that is received via at least one network;

receiving, utilizing the at least one third interface at the second node, a selection of at least one file utilizing the sixth user interface element;

sending, from the second node via the at least one network, an indication of the at least one file;

detecting, utilizing the at least one third interface at the second node, a selection of the seventh user interface element;

sending, from the second node via the at least one network, an indication of the selection of the seventh user interface element;

after sending the indication of the selection of the seventh user interface element, receiving, at the first node, a second message that includes an eighth user interface element, and that does not include a file attachment with the second message;

detecting, at the first node, a selection of the eighth user interface element of the second message;

sending, from the first node via the at least one network, an indication of the selection of the eighth user interface element of the second message;

after sending the indication of a selection of the eighth user interface element of the second message, displaying, at the first node, at least one fourth interface with a reference to the at least one file;

detecting, utilizing the at least one fourth interface at the first node, a selection of the reference to the at least one file; and

providing access, at the first node, to the at least one file, utilizing the reference.

2. The method of claim 1, wherein the at least one first interface and the at least one second interface are separate windows displayed utilizing a single web page.

3. The method of claim 1, wherein the first message is received utilizing a simple mail transfer protocol (SMTP) and the second message is received utilizing a hypertext transfer protocol (HTTP).

4. The method of claim 1, wherein the first message and the second message are both received utilizing a simple mail transfer protocol (SMTP).

5. The method of claim 1, wherein the second message is at least partially pre-written and is automatically generated without user involvement after the indication of the selection of the seventh user interface element is sent.

6. The method of claim 1, wherein the at least one of the date or the time of the indication of the at least one of the date or the time is a criterion, and access is provided to the at least one file based on whether the criterion is met.

7. The method of claim 1, wherein the file description is capable of being received in response to a first single user input, and the indication of the at least one of the

date or the time is capable of being received in response to a second single user input that immediately follows the first single user input.

8. The method of claim 1, wherein the second message does not include the file attachment, for avoiding the at least one file from having to be communicated to and stored at the first node until the communication and the storage is initiated by a user of the first node.

9. The method of claim 1, wherein a first user of the first node is capable of requesting the at least one file and accessing the at least one file, by permitting a second user of the second node to store the at least one file at at least one server that is accessible by the first user of the first node, without requiring the at least one file to be communicated from the second node to the first node as a message attachment.

10. An apparatus, comprising:

a first node configured to:

display, at the first node, a first user interface element, a second user interface element, and a third user interface element;

receive, at the first node, text associated with one or more desired files via the first user interface element that includes a textbox;

send, from the first node via at least one network, the text associated with the one or more desired files;

receive, at the first node, an object associated with at least one email address via the second user interface element;

send, from the first node via the at least one network, the object associated with the at least one email address;

detect, at the first node, a selection of the third user interface element; and

send, from the first node via the at least one network, a message associated with the selection of the third user interface element;

said first node further configured to communicate with a second node that is configured to:

after the message associated with the selection of the third user interface element is sent and based on the text associated with the one or more desired files and the object associated with the at least one email address; receive, at the second node via the at least one network, a first message including a fourth user interface element and at least one sentence including the text associated with the one or more desired files;

detect, at the second node, a selection of the fourth user interface element of the first message;

send, from the second node via the at least one network, a message associated with the selection of the fourth user interface element of the first message;

after the message associated with the selection of the fourth user interface element is sent, display, at the second node, a fifth user interface element and a sixth user interface element;

detect, at the second node, a selection of at least one file;

send, from the second node via the at least one network, a message associated with the at least one file;

detect, at the second node, a selection of the sixth user interface element; and

send, from the second node via the at least one network, a message associated with the selection of the sixth user interface element;

said first node further configured to:

after the message associated with the selection of the sixth user interface element is sent, receive, at the first node, a second message that includes a seventh user interface element, and that does not include a file attachment with the second message;

detect, at the first node, a selection of the seventh user interface element of the second message;

send, from the first node via the at least one network, a message associated with the selection of the seventh user interface element of the second message;

after the message associated with the selection of the seventh user interface element of the second message is sent, display, at the first node, an eighth user interface element associated with a reference to the at least one file; and

provide access, at the first node, to the at least one file utilizing the reference.

11. The apparatus of claim 10, wherein the first node is configured such that:

the first message and the second message are emails;

the object is the at least one email address or an alias associated with the at least one email address;

the second message is at least partially pre-written and is automatically generated without user involvement after the message associated with the selection of the sixth user interface element is sent;

the at least one file is conditionally shared based on whether a date-related criterion or a time-related criterion is met;

the second message does not include the file attachment, for avoiding the at least one file from being communicated to and stored at the first node until the communication of the at least one file is initiated by a first user of the first node via at at least one server that stores the at least one file.

12. An apparatus, comprising:

at least one non-transitory memory storing instructions; and

one or more processors in communication with the at least one non-transitory memory, wherein the one or more processors execute the instructions to:

cause, at a first node, display of at least one first interface with a first user interface element and a second user interface element, utilizing first hypertext markup language-equipped code that is sent via at least one network;

receive, from the first node via the at least one network, a file description via the first user interface element that includes a textbox, utilizing the at least one first interface;

receive, from the first node via the at least one network, an indication of at least one of a date or a time via the second user interface element, utilizing the at least one first interface;

cause, at the first node, display of at least one second interface with a third user interface element and a fourth user interface element, utilizing second hypertext markup language-equipped code that is sent via the at least one network;

receive, from the first node via the at least one network, indicia associated with at least one email address via the third user interface element, utilizing the at least one second interface;

receive, from the first node via the at least one network, an indication of a selection of the fourth user interface element, utilizing the at least one second interface;

in response to the receipt of the indication of the selection of the fourth user interface element and based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; generate a first message indicating the at least one of the date or the time, and including a fifth user interface element and at least one sentence including the file description;

send, to a second node via the at least one network, the first message;

receive, from the second node via the at least one network, an indication of a selection of the fifth user interface element of the first message;

in response to the receipt of the indication of the selection of the fifth user interface element, cause, at the second node, display of at least one third interface with a sixth user interface element and a seventh user interface element, utilizing third hypertext markup language-equipped code that is sent via at least one network;

receive, from the second node via the at least one network, an indication of a selection at least one file utilizing the sixth user interface element, utilizing the at least one third interface;

receive, from the second node via the at least one network, an indication of a selection of the seventh user interface element, utilizing the at least one third interface;

in response to the receipt of the indication of the selection of the seventh user interface element, generate a second message that includes an eighth user interface element, and that does not include a file attachment with the second message;

send, to the first node via the at least one network, the second message;

receive, from the first node via the at least one network, an indication of a selection of the eighth user interface element of the second message;

in response to the receipt of the indication of the selection of the eighth user interface element of the second message, cause, at the first node, display of at least one fourth interface with a reference to the at least one file for allowing access to the at least one file at the first node.

13. The apparatus of claim 12, wherein the apparatus is configured such that the indicia associated with the at least one email address includes an alias associated with the at least one email address.

14. The apparatus of claim 12, wherein the apparatus is configured such that the indicia associated with the at least one email address includes the at least one email address.

15. The apparatus of claim 12, wherein the apparatus is configured such that the at least one first interface and the at least one second interface are displayed utilizing a single web page.

16. The apparatus of claim 12, wherein the apparatus is configured such that the at least one first interface and the at least one second interface are separate windows displayed utilizing a single web page.

17. The apparatus of claim 12, wherein the apparatus is configured such that the first message is sent utilizing a simple mail transfer protocol (SMTP) and the second message is sent utilizing a hypertext transfer protocol (HTTP).

18. The apparatus of claim 12, wherein the apparatus is configured such that the first message and the second message are both sent utilizing a simple mail transfer protocol (SMTP).

19. The apparatus of claim 12, wherein the apparatus is configured such that the second message is at least partially pre-written and is automatically generated without user involvement in response to the receipt of the indication of the selection of the seventh user interface element.

20. The apparatus of claim 12, wherein the apparatus is configured such that the at least one of the date or the time of the indication of the at least one of the date or the time is a criterion, and the indication of the at least one file is received based on whether the criterion is met.

21. The apparatus of claim 12, wherein the apparatus is configured such that the file description is capable of being received in response to a first single user input, and the indication of the at least one of the date or the time is capable of being received in response to a second single user input that immediately follows the first single user input.

22. The apparatus of claim 12, wherein the apparatus is configured such that the second message does not include the file attachment, for avoiding the at least

one file from having to be communicated to and stored at the first node until the communication and the storage is initiated by a user of the first node.

23. The apparatus of claim 12, wherein the apparatus is configured such that a first user of the first node is capable of requesting the at least one file and accessing the at least one file, by permitting a second user of the second node to store the at least one file at at least one server that is accessible by the first user of the first node, without requiring the at least one file to be communicated from the second node to the first node as a message attachment.

24. The apparatus of claim 12, wherein the apparatus is configured such that the display of at least one fourth interface is caused utilizing fourth hypertext markup language-equipped code that is sent via at least one network, and the one or more processors execute the instructions to:

receive, from the first node via the at least one network, an indication of a selection of the reference to the at least one file, utilizing the at least one fourth interface; and

in response to the receipt of the indication of the selection of the indication of a selection of the reference to the at least one file, allow access, at the first node, to the at least one file.

25. The apparatus of claim 24, wherein the apparatus is configured such that at least one of:

said at least one first interface is an email interface;

said at least one second interface is an email interface;

said at least one third interface is an email interface;

said at least one fourth interface is an email interface;

said at least one first interface is a single interface;

said at least one second interface is a single interface;

said at least one third interface is a single interface;

said at least one fourth interface is a single interface;

said at least one first interface is an interface portion;

said at least one second interface is an interface portion;

said at least one third interface is an interface portion;

said at least one fourth interface is an interface portion;

said at least one first interface is an interface element;

said at least one second interface is an interface element;

said at least one third interface is an interface element;

said at least one fourth interface is an interface element;

said at least one first interface is different from the said at least one second interface;

said at least one first interface include a plurality of first interfaces;

said at least one second interface includes a plurality of second interfaces;

said at least one third interface includes a plurality of first interfaces;

said at least one fourth interface includes a plurality of second interfaces;

said display of the at least one first interface is caused by sending at least one message via the at least one network;

said display of the at least one first interface is caused by sending a communication including a message via the at least one network;

said display of the at least one first interface is caused by at least one inter-process message;

said display of the at least one first interface is caused by serving the at least one first interface via the at least one network;

said display of the at least one second interface is caused by sending at least one message via the at least one network;

said display of the at least one second interface is caused by sending a communication including a message via the at least one network;

said display of the at least one second interface is caused by at least one inter-process message;

said display of the at least one second interface is caused by serving the at least one first interface via the at least one network;

said display of the at least one third interface is caused by sending at least one message via the at least one network;

said display of the at least one third interface is caused by sending a communication including a message via the at least one network;

said display of the at least one third interface is caused by at least one inter-process message;

said display of the at least one third interface is caused by serving the at least one third interface via the at least one network;

said display of the at least one fourth interface is caused by sending at least one message via the at least one network;

said display of the at least one fourth interface is caused by sending a communication including a message via the at least one network;

said display of the at least one fourth interface is caused by at least one inter-process message;

said display of the at least one fourth interface is caused by serving the at least one fourth interface via the at least one network;

said first hypertext markup language-equipped code is included in a first web page, the second hypertext markup language-equipped code is included in a second web page, the third hypertext markup language-equipped code is included in a third web page, and the fourth hypertext markup language-equipped code is included in a fourth web page;

said at least one first interface includes a user interface for a keyword based schema;

said first user interface element includes a match textbox user interface element for allowing a user to enter a keyword expression;

said at least one first interface includes a checkbox;

said second user interface element includes a match user interface element for allowing a user to enter a keyword expression;

said first user interface element and said second user interface element include sub-elements of a single user interface element;

said file description includes a criterion;

said file description includes one or more words;

said at least one of the date or the time includes a criterion;

said at least one of the date or the time includes the date;

said at least one of the date or the time includes the time;

said at least one of the date or the time includes the date and the time;

said at least one of the date or the time includes a textual description of at least one of the date or the time;

said at least one of the date or the time includes a selection of at least one of the date or the time;

said third user interface element includes a menu;

said third user interface element includes a textbox;

said third user interface element includes list box;

said third user interface element includes a drop-down lists;

said indicia associated with the at least one email is generated based on a user selection of the at least one email address;

said indicia associated with the at least one email is generated based on a user locating the at least one email address;

said indicia associated with the at least one email is generated based on a user input in connection with an email-related user interface element;

said indicia associated with the at least one email is generated based on a user entry of text associated with the at least email;

said indicia includes a text string that is associated with the at least one email address;

said fourth user interface element, the fifth user interface element, the seventh user interface element, the eighth user interface element, each include different buttons;

said fourth user interface element, the fifth user interface element, the seventh user interface element, the eighth user interface element, each include different controls;

said fourth user interface element includes a send icon;

said fifth user interface element includes a data object identification response icon;

said seventh user interface element includes an access button;

said seventh user interface element includes a reply button;

said eighth user interface element includes a context menu;

said eighth user interface element includes a get command option;

said eighth user interface element includes an access button;

said eighth user interface element includes a link;

said reference includes a uniform resource identifier;

said reference includes a link;

said indication of the selection of the fourth user interface element, the indication of the selection of the fifth user interface element, the indication of the selection of the seventh user interface element, the indication of the selection of the eighth user interface element, each include a signal;

said indication of the selection of the fourth user interface element, the indication of the selection of the fifth user interface element, the indication of the selection of the seventh user interface element, the indication of the selection of the eighth user interface element, each include a message;

said indication of the selection of the fourth user interface element, the indication of the selection of the fifth user interface element, the indication of the selection of the seventh user interface element, the indication of the selection of the eighth user interface element, each result from user input received via at least one input device in communication with the one or more processors;

said indication of the selection of the fourth user interface element, the indication of the selection of the fifth user interface element, the indication of the selection of the seventh user interface element, the indication of the selection of the eighth user interface element, each result from a selection of a reference associated with a corresponding user interface element;

said sixth user interface element prompts is capable of being utilized by causing a display of a file explorer interface;

said sixth user interface element prompts is capable of being utilized by causing a display of an interface for use in selecting the at least one file;

said sixth user interface element includes a list of a plurality of files;

said sixth user interface element includes a button control;

said sixth user interface element includes an access button;

said indication of the selection of the at least one file results completely from a manual operation;

said indication of the selection of the at least one file results, at least in part, from a manual operation;

said indication of the selection of the at least one file results from an automated operation;

said indication of the selection of the at least one file results in a communication of the at least one file over the at least one network;

said indication of the selection of the at least one file includes the at least one file itself;

said indication of the selection of the at least one file includes receipt of information associated with the at least one file;

said first message includes a simple mail transfer protocol (SMTP) message, a hypertext transfer protocol (HTTP) message, a file transfer protocol (FTP) message, one or more transmission control protocol (TCP) data units, one or more IP packets, a; email message, or a short message service message;

said second message includes a simple mail transfer protocol (SMTP) message, a hypertext transfer protocol (HTTP) message, a file transfer protocol (FTP) message, one or more transmission control protocol (TCP) data units, one or more IP packets, a; email message, or a short message service message;

said first message and said second message are transmitted via the same protocol;

said first message and said second message are transmitted via different protocols;

said first email is generated based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address, by being generated in response to the receipt of all of the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; or

said first email is generated based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address, by being generated only after the receipt of all the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address.

26. The apparatus of claim 12, wherein the apparatus is configured such that, in response to the receipt of the indication of the selection of the seventh user interface element, a creation of a representation of the at least one file is caused, at the first node, in a location among one or more folders on a file explorer interface that is not a web page and is displayed utilizing a client-based file explorer application, such that the at least one file is accessible via both the second message and the file explorer interface.

27. The apparatus of claim 26, wherein the apparatus is configured such that the second message is sent without including the at least one file as an attachment of the second message, and a storage at the first node does not store the at least one file when the creation of the representation of the at least one file is caused, both for avoiding the at least one file from having to be communicated to and stored at the first node until the at least one file is requested by a user at the first node using the representation or the reference.

28. An apparatus, comprising:

at least one non-transitory memory storing instructions; and

one or more processors in communication with the at least one non-transitory memory, wherein the one or more processors execute the instructions to:

send a first communication over at least one network that results in display, at a first node, of a first user interface element, a second user interface element, and a third user interface element;

receive, from the first node via the at least one network, text associated with one or more desired files via the first user interface element that includes a textbox;

receive, from the first node via the at least one network, an object associated with at least one email address via the second user interface element;

receive, from the first node via the at least one network, a message associated with a selection of the third user interface element;

in response to the receipt of the message associated with the selection of the third user interface element and based on the text associated with the one or more desired files and the object associated with the at least one email address;

generate a first message including a fourth user interface element and the text associated with the one or more desired files;

send, to a second node via the at least one network, the first message;

receive, from the second node via the at least one network, a message associated with a selection of the fourth user interface element of the first message;

in response to the receipt of the message associated with the selection of the fourth user interface element, send a second communication over the at least one network that results in display, at the second node, of a fifth user interface element;

receive, from the second node via the at least one network, a message associated with a selection at least one file;

receive, from the second node via the at least one network, a message associated with a selection of the fifth user interface element;

in response to the receipt of the message associated with the selection of the fifth user interface element, generate a second message that includes an sixth user interface element, and that does not include a file attachment with the second message;

send, to the first node via the at least one network, the second message;

receive, from the first node via the at least one network, a message associated with a selection of the sixth user interface element of the second message;

in response to the receipt of the message associated with the selection of the sixth user interface element of the second message, send a third communication over the at least one network that results in display, at the first node, of a reference to the at least one file; and

allow access, at the first node, to the at least one file via the reference.

29. The apparatus of claim 28, wherein the apparatus is configured such that:

the first message and the second message are email messages;

the first communication, the second communication, and the third communication each include hypertext markup language-equipped code for being displayed via at least one web page;

the object includes the at least one email address or an alias associated with the at least one email address;

the second message is at least partially pre-written and is automatically generated without user involvement after the receipt of the message associated with the selection of the fifth user interface element; and

the message associated with the at least one file is conditionally received based on whether a date-related criterion or a time-related criterion is met.

30. The apparatus of claim 29, wherein the apparatus is configured such that, in response to the receipt of the indication of the selection of the fifth user interface element, a creation of a representation of the at least one file is caused, at the first node, in a location among one or more folders on a file explorer interface that is not a web page and is displayed utilizing a client-based file explorer application, such that the at least one file is accessible via both the second message and the file explorer interface, wherein the apparatus is further configured such that the second message is sent without including the at least

one file as an attachment of the second message, and a storage at the first node does not store the at least one file when the creation of the representation of the at least one file is caused, both for avoiding the at least one file from having to be communicated to and stored at the first node until the at least one file is requested by a user at the first node using the representation or the reference.

ABSTRACT

Methods, systems, and apparatuses are provided that are configured to: display, at a first node, at least one first interface with a first user interface element and a second user interface element; display, at the first node, at least one second interface; generate a first message; display, at the second node, at least one third interface; generate a second message that does not include a file attachment with the second message; display, at the first node, at least one fourth interface with a reference to at least one file; and allow access, at the first node, to the at least one file.

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION
First Named Inventor/Applicant Name:	Robert Paul Morris
Filer:	Patrick Edgar Caldwell
Attorney Docket Number:	DR0149A-CIP

Filed as Small Entity

Filing Fees for Track I Prioritized Examination - Nonprovisional Application under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
UTILITY FILING FEE (ELECTRONIC FILING)	4011	1	70	70
UTILITY SEARCH FEE	2111	1	300	300
UTILITY EXAMINATION FEE	2311	1	360	360
REQUEST FOR PRIORITIZED EXAMINATION	2817	1	2000	2000
Pages:				
UTILITY APPL SIZE FEE PER 50 SHEETS >100	2081	1	200	200
Claims:				
CLAIMS IN EXCESS OF 20	2202	10	40	400

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
INDEPENDENT CLAIMS IN EXCESS OF 3	2201	1	210	210
Miscellaneous-Filing:				
PROCESSING FEE, EXCEPT PROV. APPLS.	2830	1	70	70
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
			Total in USD ($)	**3610**

Electronic Acknowledgement Receipt

EFS ID:	30558402
Application Number:	15724223
International Application Number:	
Confirmation Number:	1049
Title of Invention:	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION
First Named Inventor/Applicant Name:	Robert Paul Morris
Customer Number:	92045
Filer:	Patrick Edgar Caldwell
Filer Authorized By:	
Attorney Docket Number:	DR0149A-CIP
Receipt Date:	03-OCT-2017
Filing Date:	
Time Stamp:	21:24:08
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$3610
RAM confirmation Number	100417INTEFSW21245200
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	DR0149A-CIP_ADS_vF_03-Oct-2017.pdf	83973 bb80595b3504dca25d6c2b402cb04b95a8ec0dd1	no	8
Warnings:					
Information:					
This is not an USPTO supplied ADS fillable form					
2	Oath or Declaration filed	DR0149A-CIP_Decl_vF_03-Oct-2017.pdf	742277 a07ce4eb553b8ae45405f8fdbf7dc1d76c4e6bcf	no	2
Warnings:					
Information:					
3	Power of Attorney	DR0149A-CIP_POA_vF_03-Oct-2017.pdf	106814 31031a428e184c05516ae914efa5c486f8a54677	no	2
Warnings:					
Information:					
4	TrackOne Request	DR0149A-CIP_Track1_vF_03-Oct-2017.pdf	83491 777498a38b7e72a3d8a0106b5e84acf9d7eaeadd	no	1
Warnings:					
Information:					
5	Drawings-only black and white line drawings	DR0149A-CIP_Figs_vF_03-Oct-2017.pdf	1712023 4d6af40980fce270c85b654164c0c2ea832b7af3	no	15
Warnings:					
Information:					
6		DR0149A-CIP_Spec_vF_03-Oct-2017.pdf	459949 f3521979029ff8bcdb116794706825b273c1caa5	yes	111

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Specification	1	74
	Claims	75	110
	Abstract	111	111

Warnings:					
Information:					
7	Fee Worksheet (SB06)	fee-info.pdf	43796 590f623db6655ff18c52449c4c0f2e39cec547a5	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	3232323		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149A-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2:

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Robert	Paul	Morris	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	Raleigh	State/Province	NC	Country of Residence	US

Mailing Address of Inventor:

Address 1	712 Latta Street		
Address 2			
City	Raleigh	State/Province	NC
Postal Code	27607	Country	US

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	92045		
Email Address	pcaldwell@thecaldwellfirm.com	Add Email	Remove Email

Application Information:

Title of the Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION		
Attorney Docket Number	DR0149A-CIP	Small Entity Status Claimed	☒
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)		Suggested Figure for Publication (if any)	

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149A-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☒ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	◉ Customer Number	○ US Patent Practitioner	○ Limited Recognition (37 CFR 11.9)
Customer Number	92045		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.
When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
	Continuation in part of	14274623	2014-05-09

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149A-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
14274623	Continuation in part of	13624906	2012-09-22
Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
14274623	Continuation in part of	13626635	2012-09-25

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button.

Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)[i] the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

			Remove
Application Number	Country[i]	Filing Date (YYYY-MM-DD)	Access Code[i] (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button.

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149A-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☒ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149A-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Applicant 1

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Clear

◉ Assignee	○ Legal Representative under 35 U.S.C. 117	○ Joint Inventor
○ Person to whom the inventor is obligated to assign.	○ Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor:

If the Applicant is an Organization check here. ☒

Organization Name	Cedar Point Partners, LLC		
Mailing Address Information For Applicant:			
Address 1	712 Latta Street		
Address 2			
City	Raleigh	**State/Province**	NC
Country	US	Postal Code	27607
Phone Number		Fax Number	
Email Address			

Additional Applicant Data may be generated within this form by selecting the Add button.

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149A-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1			
Address 2			
City		**State/Province**	
Country		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button.

Signature:

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). **However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).**

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/Patrick Caldwell/			Date (YYYY-MM-DD)	2017-10-03
First Name	Patrick	Last Name	Caldwell	Registration Number	44580

Additional Signature may be generated within this form by selecting the Add button.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149A-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1 The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3 A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent CooperationTreaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

PTO/AIA/01 (06-12)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION

As the below named inventor, I hereby declare that:

This declaration is directed to:

- [■] The attached application, or
- [ ] United States application or PCT international application number _____ filed on _____.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Robert Paul Morris Date (Optional) :_____

Signature: /Robert Paul Morris/

Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form or must have been previously filed. Use an additional PTO/AIA/01 form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 1 minute to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 92045

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

Sitting Man, LLC

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature	/Robert Paul Morris/	Date (Optional)	
Name	Robert Paul Morris		
Title	Managing Member		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☐ Total of forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

ABSTRACT

Methods, systems, and apparatuses are provided that are configured to: display, at a first node, at least one first interface with a first user interface element and a second user interface element; display, at the first node, at least one second interface; generate a first message; display, at the second node, at least one third interface; generate a second message that does not include a file attachment with the second message; display, at the first node, at least one fourth interface with a reference to at least one file; and allow access, at the first node, to the at least one file.

I CLAIM:

1. A computer-implemented method, comprising:

displaying, at a first node, at least one first interface with a first user interface element and a second user interface element, utilizing first hypertext markup language-equipped code that is received via at least one network;

receiving, utilizing the at least one first interface at the first node, a file description via the first user interface element that includes a textbox;

sending, from the first node via the at least one network, the file description;

receiving, utilizing the at least one first interface at the first node, an indication of at least one of a date or a time via the second user interface element;

sending, from the first node via the at least one network, the indication of the at least one of the date or the time;

displaying, at the first node, at least one second interface with a third user interface element and a fourth user interface element, utilizing second hypertext markup language-equipped code that is received via the at least one network;

receiving, utilizing the at least one second interface at the first node, indicia associated with at least one email address via the third user interface element;

sending, from the first node via the at least one network, the indicia associated with the at least one email address;

detecting, at the first node, a selection of the fourth user interface element;

sending, from the first node via the at least one network, an indication of the selection of the fourth user interface element;

after sending the indication of the selection of the fourth user interface element and based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; receiving, at a second node via the at least one network, a first message indicating the at least one of the date or the time, and including a fifth user interface element and at least one sentence including the file description;

detecting, at the second node, a selection of the fifth user interface element of the first message;

sending, from the second node via the at least one network, an indication of the selection of the fifth user interface element of the first message;

after sending the indication of the selection of the fifth user interface element, displaying, at the second node, at least one third interface with a sixth user interface element and a seventh user interface element, utilizing third hypertext markup language-equipped code that is received via at least one network;

receiving, utilizing the at least one third interface at the second node, a selection of at least one file utilizing the sixth user interface element;

sending, from the second node via the at least one network, an indication of the at least one file;

detecting, utilizing the at least one third interface at the second node, a selection of the seventh user interface element;

sending, from the second node via the at least one network, an indication of the selection of the seventh user interface element;

after sending the indication of the selection of the seventh user interface element, receiving, at the first node, a second message that includes an eighth user interface element, and that does not include a file attachment with the second message;

detecting, at the first node, a selection of the eighth user interface element of the second message;

sending, from the first node via the at least one network, an indication of the selection of the eighth user interface element of the second message;

after sending the indication of a selection of the eighth user interface element of the second message, displaying, at the first node, at least one fourth interface with a reference to the at least one file;

detecting, utilizing the at least one fourth interface at the first node, a selection of the reference to the at least one file; and

providing access, at the first node, to the at least one file, utilizing the reference.

2. The method of claim 1, wherein the at least one first interface and the at least one second interface are separate windows displayed utilizing a single web page.

3. The method of claim 1, wherein the first message is received utilizing a simple mail transfer protocol (SMTP) and the second message is received utilizing a hypertext transfer protocol (HTTP).

4. The method of claim 1, wherein the first message and the second message are both received utilizing a simple mail transfer protocol (SMTP).

5. The method of claim 1, wherein the second message is at least partially pre-written and is automatically generated without user involvement after the indication of the selection of the seventh user interface element is sent.

6. The method of claim 1, wherein the at least one of the date or the time of the indication of the at least one of the date or the time is a criterion, and access is provided to the at least one file based on whether the criterion is met.

7. The method of claim 1, wherein the file description is capable of being received in response to a first single user input, and the indication of the at least one of the

date or the time is capable of being received in response to a second single user input that immediately follows the first single user input.

8. The method of claim 1, wherein the second message does not include the file attachment, for avoiding the at least one file from having to be communicated to and stored at the first node until the communication and the storage is initiated by a user of the first node.

9. The method of claim 1, wherein a first user of the first node is capable of requesting the at least one file and accessing the at least one file, by permitting a second user of the second node to store the at least one file at at least one server that is accessible by the first user of the first node, without requiring the at least one file to be communicated from the second node to the first node as a message attachment.

10. An apparatus, comprising:

a first node configured to:

display, at the first node, a first user interface element, a second user interface element, and a third user interface element;

receive, at the first node, text associated with one or more desired files via the first user interface element that includes a textbox;

send, from the first node via at least one network, the text associated with the one or more desired files;

receive, at the first node, an object associated with at least one email address via the second user interface element;

send, from the first node via the at least one network, the object associated with the at least one email address;

detect, at the first node, a selection of the third user interface element; and

send, from the first node via the at least one network, a message associated with the selection of the third user interface element;

said first node further configured to communicate with a second node that is configured to:

after the message associated with the selection of the third user interface element is sent and based on the text associated with the one or more desired files and the object associated with the at least one email address; receive, at the second node via the at least one network, a first message including a fourth user interface element and at least one sentence including the text associated with the one or more desired files;

detect, at the second node, a selection of the fourth user interface element of the first message;

send, from the second node via the at least one network, a message associated with the selection of the fourth user interface element of the first message;

after the message associated with the selection of the fourth user interface element is sent, display, at the second node, a fifth user interface element and a sixth user interface element;

detect, at the second node, a selection of at least one file;

send, from the second node via the at least one network, a message associated with the at least one file;

detect, at the second node, a selection of the sixth user interface element; and

send, from the second node via the at least one network, a message associated with the selection of the sixth user interface element;

said first node further configured to:

after the message associated with the selection of the sixth user interface element is sent, receive, at the first node, a second message that includes a seventh user interface element, and that does not include a file attachment with the second message;

detect, at the first node, a selection of the seventh user interface element of the second message;

send, from the first node via the at least one network, a message associated with the selection of the seventh user interface element of the second message;

after the message associated with the selection of the seventh user interface element of the second message is sent, display, at the first node, an eighth user interface element associated with a reference to the at least one file; and

provide access, at the first node, to the at least one file utilizing the reference.

11. The apparatus of claim 10, wherein the first node is configured such that:

the first message and the second message are emails;

the object is the at least one email address or an alias associated with the at least one email address;

the second message is at least partially pre-written and is automatically generated without user involvement after the message associated with the selection of the sixth user interface element is sent;

the at least one file is conditionally shared based on whether a date-related criterion or a time-related criterion is met;

the second message does not include the file attachment, for avoiding the at least one file from being communicated to and stored at the first node until the communication of the at least one file is initiated by a first user of the first node via at at least one server that stores the at least one file.

12. An apparatus, comprising:

at least one non-transitory memory storing instructions; and

one or more processors in communication with the at least one non-transitory memory, wherein the one or more processors execute the instructions to:

cause, at a first node, display of at least one first interface with a first user interface element and a second user interface element, utilizing first hypertext markup language-equipped code that is sent via at least one network;

receive, from the first node via the at least one network, a file description via the first user interface element that includes a textbox, utilizing the at least one first interface;

receive, from the first node via the at least one network, an indication of at least one of a date or a time via the second user interface element, utilizing the at least one first interface;

cause, at the first node, display of at least one second interface with a third user interface element and a fourth user interface element, utilizing second hypertext markup language-equipped code that is sent via the at least one network;

receive, from the first node via the at least one network, indicia associated with at least one email address via the third user interface element, utilizing the at least one second interface;

receive, from the first node via the at least one network, an indication of a selection of the fourth user interface element, utilizing the at least one second interface;

in response to the receipt of the indication of the selection of the fourth user interface element and based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; generate a first message indicating the at least one of the date or the time, and including a fifth user interface element and at least one sentence including the file description;

send, to a second node via the at least one network, the first message;

receive, from the second node via the at least one network, an indication of a selection of the fifth user interface element of the first message;

in response to the receipt of the indication of the selection of the fifth user interface element, cause, at the second node, display of at least one third interface with a sixth user interface element and a seventh user interface element, utilizing third hypertext markup language-equipped code that is sent via at least one network;

receive, from the second node via the at least one network, an indication of a selection at least one file utilizing the sixth user interface element, utilizing the at least one third interface;

receive, from the second node via the at least one network, an indication of a selection of the seventh user interface element, utilizing the at least one third interface;

in response to the receipt of the indication of the selection of the seventh user interface element, generate a second message that includes an eighth user interface element, and that does not include a file attachment with the second message;

send, to the first node via the at least one network, the second message;

receive, from the first node via the at least one network, an indication of a selection of the eighth user interface element of the second message;

in response to the receipt of the indication of the selection of the eighth user interface element of the second message, cause, at the first node, display of at least one fourth interface with a reference to the at least one file for allowing access to the at least one file at the first node.

13. The apparatus of claim 12, wherein the apparatus is configured such that the indicia associated with the at least one email address includes an alias associated with the at least one email address.

14. The apparatus of claim 12, wherein the apparatus is configured such that the indicia associated with the at least one email address includes the at least one email address.

15. The apparatus of claim 12, wherein the apparatus is configured such that the at least one first interface and the at least one second interface are displayed utilizing a single web page.

16. The apparatus of claim 12, wherein the apparatus is configured such that the at least one first interface and the at least one second interface are separate windows displayed utilizing a single web page.

17. The apparatus of claim 12, wherein the apparatus is configured such that the first message is sent utilizing a simple mail transfer protocol (SMTP) and the second message is sent utilizing a hypertext transfer protocol (HTTP).

18. The apparatus of claim 12, wherein the apparatus is configured such that the first message and the second message are both sent utilizing a simple mail transfer protocol (SMTP).

19. The apparatus of claim 12, wherein the apparatus is configured such that the second message is at least partially pre-written and is automatically generated without user involvement in response to the receipt of the indication of the selection of the seventh user interface element.

20. The apparatus of claim 12, wherein the apparatus is configured such that the at least one of the date or the time of the indication of the at least one of the date or the time is a criterion, and the indication of the at least one file is received based on whether the criterion is met.

METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION

RELATED APPLICATIONS

[0001] The present application is a continuation of and claims priority to U.S. Patent Application No. 15/724,223 filed 10-03-2017 and entitled “METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION” which is a continuation-in-part of and claims priority to U.S. Patent Application No. 14/274,623 (published US 2014-0365588 A1) filed 05/09/2014 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION," which, in turn, is a continuation-in-part of and claims priority to U.S. Patent Application No. 13/624,906 (published US 2014-0089419 A1) filed 09/22/2012 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION" and U.S. Patent Application No. 13/626,635 (published under US 2014-0089421 A1) filed 09/25/2012 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR SHARING A DATA OBJECT IN A DATA STORE VIA A COMMUNICATION."

[0002] This application is related to the following U.S. Patent Applications:

[0003] U.S. Patent Application No. 12/833,016 (published US 2012-0011444 A1) filed on 2010/07/09, entitled “METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR REFERENCING AN ATTACHMENT IN A COMMUNICATION”;

[0004] U.S. Patent Application No. 12/833,014 (published US 2012-0011207 A1) filed on 2010/07/09, entitled “METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A REQUEST FOR A RESOURCE IN A COMMUNICATION”;

[0005] U.S. Patent Application No. 13/624,906 (published US 2014-0089419 A1) filed 2012/09/22 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR

PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION;" AND

[0006] U.S. Patent Application No. 13/626,635 (published under US 2014-0089421 A1) filed 2012/09/25 entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR SHARING A DATA OBJECT IN A DATA STORE VIA A COMMUNICATION;" the entire contents of each are herein incorporated by reference for all purposes.

BACKGROUND

[0007] While receiving attachments in a network communication, such as email, is common, most attachments received are unsolicited or unrequested. To request an attachment, a user typically sends the request as voice and/or text data in a voice and/or text message heard and/or read by another user. The request may be as vague or as specific as the language used by the requesting user. The other user must interpret the request and find a file or other data object that seems to match the request. The other user in many cases locates a “matching” resource using a program or application other than the communications agent that received the request. For example, a search program may be used to search a hard-drive by the other user. The other user must enter the search criteria.

[0008] Accordingly, there exists a need for methods, systems, and computer program products for processing a data object identification request in a communication. .

SUMMARY

[0009] The following presents a simplified summary of the disclosure in order to provide a basic understanding to the reader. This summary is not an extensive overview of the disclosure and it does not identify key/critical elements of the invention or delineate the scope of the invention. Its sole purpose is to present some concepts disclosed herein in a simplified form as a prelude to the more detailed description that is presented later.

[0010] In one embodiment, an apparatus is provided, comprising: at least one non-transitory memory storing instructions; and one or more processors in communication with the at least one non-transitory memory, wherein the one or more processors execute the instructions to: cause, at a first node, display of at least one first interface with a first user interface element and a second user interface element, utilizing first hypertext markup language-equipped code that is sent via at least one network; receive, from the first node via the at least one network, a file description via the first user interface element that includes a textbox, utilizing the at least one first interface; receive, from the first node via the at least one network, an indication of at least one of a date or a time via the second user interface element, utilizing the at least one first interface; cause, at the first node, display of at least one second interface with a third user interface element and a fourth user interface element, utilizing second hypertext markup language-equipped code that is sent via the at least one network; receive, from the first node via the at least one network, indicia associated with at least one email address via the third user interface element, utilizing the at least one second interface; receive, from the first node via the at least one network, an indication of a selection of the fourth user interface element, utilizing the at least one second interface; in response to the receipt of the indication of the selection of the fourth user interface element and based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; generate a first message indicating the at least one of the date or the time, and including a fifth user interface element and at least one sentence including the file description; send, to a second node via the at least one network, the first message; receive, from the second node via the at least one

network, an indication of a selection of the fifth user interface element of the first message; in response to the receipt of the indication of the selection of the fifth user interface element, cause, at the second node, display of at least one third interface with a sixth user interface element and a seventh user interface element, utilizing third hypertext markup language-equipped code that is sent via at least one network; receive, from the second node via the at least one network, an indication of a selection at least one file utilizing the sixth user interface element, utilizing the at least one third interface; receive, from the second node via the at least one network, an indication of a selection of the seventh user interface element, utilizing the at least one third interface; in response to the receipt of the indication of the selection of the seventh user interface element, generate a second message that includes an eighth user interface element, and that does not include a file attachment with the second message; send, to the first node via the at least one network, the second message; receive, from the first node via the at least one network, an indication of a selection of the eighth user interface element of the second message; in response to the receipt of the indication of a selection of the eighth user interface element of the second message, cause, at the first node, display of at least one fourth interface with a reference to the at least one file for allowing access, at the first node, to the at least one file.

[0011] In another embodiment, an apparatus is provided, comprising at least one non-transitory memory storing instructions; and one or more processors in communication with the at least one non-transitory memory, wherein the one or more processors execute the instructions to: send a first communication over at least one network that results in display, at a first node, of a first user interface element, a second user interface element, and a third user interface element; receive, from the first node via the at least one network, text associated with one or more desired files via the first user interface element that includes a textbox; receive, from the first node via the at least one network, an object associated with at least one email address via the second user interface element; receive, from the first node via the at least one network, a message associated with a selection of the third user interface element; in response to the receipt of the message associated with the selection of the third user interface element and based on the text associated with the one or more desired files and the object

associated with the at least one email address; generate a first message including a fourth user interface element and the text associated with the one or more desired files; send, to a second node via the at least one network, the first message; receive, from the second node via the at least one network, a message associated with a selection of the fourth user interface element of the first message; in response to the receipt of the message associated with the selection of the fourth user interface element, send a third communication over the at least one network that results in display, at the second node, of a fifth user interface element; receive, from the second node via the at least one network, a message associated with a selection at least one file; receive, from the second node via the at least one network, a message associated with a selection of the fifth user interface element; in response to the receipt of the message associated with the selection of the fifth user interface element, generate a second message that includes an sixth user interface element, and that does not include a file attachment with the second message; send, to the first node via the at least one network, the second message; receive, from the first node via the at least one network, a message associated with a selection of the sixth user interface element of the second message; in response to the receipt of the message associated with the selection of the sixth user interface element of the second message, send a fourth communication over the at least one network that results in display, at the first node, of a reference to the at least one file; and allow access, at the first node, to the at least one file via the reference.

[0012] In yet another embodiment, a computer-implemented method is provided, comprising: displaying, at a first node, at least one first interface with a first user interface element and a second user interface element, utilizing first hypertext markup language-equipped code that is received via at least one network; receiving, utilizing the at least one first interface at the first node, a file description via the first user interface element that includes a textbox; sending, from the first node via the at least one network, the file description; receiving, utilizing the at least one first interface at the first node, an indication of at least one of a date or a time via the second user interface element; sending, from the first node via the at least one network, the indication of the at least one of the date or the time; displaying, at the first node, at least one second interface with a third user interface element and a fourth user interface element, utilizing

second hypertext markup language-equipped code that is received via the at least one network; receiving, utilizing the at least one second interface at the first node, indicia associated with at least one email address via the third user interface element; sending, from the first node via the at least one network, the indicia associated with the at least one email address; detecting, at the first node, a selection of the fourth user interface element; sending, from the first node via the at least one network, an indication of the selection of the fourth user interface element; after sending the indication of the selection of the fourth user interface element and based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; receiving, at a second node via the at least one network, a first message indicating the at least one of the date or the time, and including a fifth user interface element and at least one sentence including the file description; detecting, at the second node, a selection of the fifth user interface element of the first message; sending, from the second node via the at least one network, an indication of the selection of the fifth user interface element of the first message; after sending the indication of the selection of the fifth user interface element, displaying, at the second node, at least one third interface with a sixth user interface element and a seventh user interface element, utilizing third hypertext markup language-equipped code that is received via at least one network; receiving, utilizing the at least one third interface at the second node, a selection of at least one file utilizing the sixth user interface element; sending, from the second node via the at least one network, an indication of the at least one file; detecting, utilizing the at least one third interface at the second node, a selection of the seventh user interface element; sending, from the second node via the at least one network, an indication of the selection of the seventh user interface element; after sending the indication of the selection of the seventh user interface element, receiving, at the first node, a second message that includes an eighth user interface element, and that does not include a file attachment with the second message; detecting, at the first node, a selection of the eighth user interface element of the second message; sending, from the first node via the at least one network, an indication of the selection of the eighth user interface element of the second message; after sending the indication of a selection of the eighth user interface element of the second

message, displaying, at the first node, at least one fourth interface with a reference to the at least one file; detecting, utilizing the at least one fourth interface at the first node, a selection of the reference to the at least one file; and providing access, at the first node, to the at least one file, utilizing the reference.

[0013] In still yet another embodiment, an apparatus is provided, comprising: a first node configured to: display, at the first node, a first user interface element, a second user interface element, and a third user interface element; receive, at the first node, text associated with one or more desired files via the first user interface element that includes a textbox; send, from the first node via at least one network, the text associated with the one or more desired files; receive, at the first node, an object associated with at least one email address via the second user interface element; send, from the first node via the at least one network, the object associated with the at least one email address; detect, at the first node, a selection of the third user interface element; and send, from the first node via the at least one network, a message associated with the selection of the third user interface element; said first node further configured to communicate with a second node that is configured to: after the message associated with the selection of the third user interface element is sent and based on the text associated with the one or more desired files and the object associated with the at least one email address; receive, at the second node via the at least one network, a first message including a fourth user interface element and at least one sentence including the text associated with the one or more desired files; detect, at the second node, a selection of the fourth user interface element of the first message; send, from the second node via the at least one network, a message associated with the selection of the fourth user interface element of the first message; after the message associated with the selection of the fourth user interface element is sent, display, at the second node, a fifth user interface element and a sixth user interface element; detect, at the second node, a selection of at least one file; send, from the second node via the at least one network, a message associated with the at least one file; detect, at the second node, a selection of the sixth user interface element; and send, from the second node via the at least one network, a message associated with the selection of the sixth user interface element; said first node further configured to: after the message associated

with the selection of the sixth user interface element is sent, receive, at the first node, a second message that includes a seventh user interface element, and that does not include a file attachment with the second message; detect, at the first node, a selection of the seventh user interface element of the second message; send, from the first node via the at least one network, a message associated with the selection of the seventh user interface element of the second message; after the message associated with the selection of the seventh user interface element of the second message is sent, display, at the first node, an eighth user interface element associated with a reference to the at least one file; and provide access, at the first node, to the at least one file utilizing the reference.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] Objects and advantages of the present invention will become apparent to those skilled in the art upon reading this description in conjunction with the accompanying drawings, in which like reference numerals have been used to designate like or analogous elements, and in which:

[0015] FIG. **1** is a block diagram illustrating an exemplary hardware device included in and/or otherwise providing an execution environment in which the subject matter may be implemented;

[0016] FIG. **2A** is a flow diagram illustrating a method for processing a data object identification request in a communication according to an aspect of the subject matter described herein;

[0017] FIG. **2B** is a flow diagram illustrating a method for processing a data object identification request in a communication according to an aspect of the subject matter described herein;

[0018] FIG. **3A** is a block diagram illustrating an arrangement of components for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0019] FIG. **3B** is a block diagram illustrating an arrangement of components for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0020] FIG. **4** is a block diagram illustrating an arrangement of components for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0021] FIG. **5** is a network diagram illustrating a system for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0022] FIG. **6A** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0023] FIG. **6B** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0024] FIG. **6C** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0025] FIG. **6D** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0026] FIG. **6E** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0027] FIG. **6F** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0028] FIG. **6G** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0029] FIG. **7** is a message flow diagram illustrating an exemplary data and execution flow for processing a data object identification request in a communication according to an aspect of the subject matter described herein;

[0030] FIG. **8A** illustrates an exemplary portion of a communication between a first execution environment and a second execution environment according to an aspect of the subject matter described herein; and

[0031] FIG. **8B** illustrates another exemplary portion of a communication between a first execution environment and a second execution environment according to an aspect of the subject matter described herein.

DETAILED DESCRIPTION

[0032] One or more aspects of the disclosure are described with reference to the drawings, wherein like reference numerals are generally utilized to refer to like elements throughout, and wherein the various structures are not necessarily drawn to scale. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of one or more aspects of the disclosure. It may be evident, however, to one skilled in the art, that one or more aspects of the disclosure may be practiced with a lesser degree of these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate describing one or more aspects of the disclosure.

[0033] Unless otherwise defined, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this disclosure belongs. Although methods, components, and devices similar or equivalent to those described herein can be used in the practice or testing of the subject matter described herein, suitable methods, components, and devices are described below.

[0034] All publications, patent applications, patents, and other references mentioned herein are incorporated by reference in their entirety. In case of conflict, the present specification, including definitions, will control. In addition, the materials, methods, and examples are illustrative only and not intended to be limiting.

[0035] An exemplary device included in an execution environment that may be configured according to the subject matter is illustrated in FIG. **1**. An "execution environment", as used herein, is an arrangement of hardware and, in some aspects, software that may be further configured to include and/or otherwise host an arrangement of components for performing a method of the subject matter described herein. An execution environment includes and/or is otherwise provided by one or more devices. The execution environment is said to be the execution environment of the device and/or devices. An execution environment may be and/or may include a virtual execution environment including software components operating in a host execution environment. Exemplary devices included in and/or otherwise providing suitable execution environments for configuring according to the subject matter include personal computers, notebook computers, tablet computers, servers, portable electronic devices, handheld electronic devices, mobile devices, multiprocessor devices, distributed systems, consumer electronic devices, routers, communication servers, and/or any other suitable devices. Those skilled in the art will understand that the components illustrated in FIG. **1** are exemplary and may vary by particular execution environment.

[0036] FIG. **1** illustrates a hardware device **100** included in an execution environment **102**. FIG. **1** illustrates that execution environment **102** includes a processor **104**, such as one or more microprocessors; a physical processor memory **106** including storage locations identified by addresses in a physical memory address space of processor **104**; a persistent secondary storage **108**, such as one or more hard drives and/or flash

storage media; an input device adapter **110**, such as a key or keypad hardware, a keyboard adapter, and/or a mouse adapter; an output device adapter **112**, such as a display and/or an audio adapter for presenting information to a user; a network interface component, illustrated by a network interface adapter **114**, for communicating via a network such as a LAN and/or WAN; and a communication mechanism that operatively couples elements **104-114**, illustrated as a bus **116**. Elements **104-114** may be operatively coupled by various means. Bus **116** may comprise any type of bus architecture, including a memory bus, a peripheral bus, a local bus, and/or a switching fabric.

[0037] As used herein a "processor" is an instruction execution machine, apparatus, or device. A processor may include one or more electrical, optical, and/or mechanical components that operate in interpreting and executing program instructions. Exemplary processors include one or more microprocessors, digital signal processors (DSPs), graphics processing units, application-specific integrated circuits (ASICs), optical or photonic processors, and/or field programmable gate arrays (FPGAs). Processor **104** may access machine code instructions and data via one or more memory address spaces in addition to the physical memory address space. A memory address space includes addresses identifying locations in a processor memory. The addresses in a memory address space are included in defining a processor memory. Processor **104** may have more than one processor memory. Thus, processor **104** may have more than one memory address space. Processor **104** may access a location in a processor memory by processing an address identifying the location. The processed address may be identified by an operand of a machine code instruction and/or may be identified by a register or other portion of processor **104**.

[0038] FIG. **1** illustrates a virtual processor memory **118** spanning at least part of physical processor memory **106** and may span at least part of persistent secondary storage **108**. Virtual memory addresses in a memory address space may be mapped to physical memory addresses identifying locations in physical processor memory **106**. An address space for identifying locations in a virtual processor memory is referred to as a virtual memory address space; its addresses are referred to as virtual memory addresses; and its processor memory is referred to as a virtual processor memory or

virtual memory. The term "processor memory" may refer to physical processor memory, such as processor memory **106**, and/or may refer to virtual processor memory, such as virtual processor memory **118**, depending on the context in which the term is used.

[0039] Physical processor memory **106** may include various types of memory technologies. Exemplary memory technologies include static random access memory (SRAM) and/or dynamic RAM (DRAM) including variants such as dual data rate synchronous DRAM (DDR SDRAM), error correcting code synchronous DRAM (ECC SDRAM), RAMBUS DRAM (RDRAM), and/or XDR™ DRAM. Physical processor memory **106** may include volatile memory as illustrated in the previous sentence and/or may include nonvolatile memory such as nonvolatile flash RAM (NVRAM) and/or ROM.

[0040] Persistent secondary storage **108** may include one or more flash memory storage devices, one or more hard disk drives, one or more magnetic disk drives, and/or one or more optical disk drives. Persistent secondary storage may include a removable data storage medium. The drives and their associated tangible computer readable storage media provide volatile and/or nonvolatile storage for computer-readable instructions, data structures, program components, and other data for execution environment **102**.

[0041] Execution environment **102** may include software components stored in persistent secondary storage **108**, in remote storage accessible via a network, and/or in a processor memory. FIG. **1** illustrates execution environment **102** including an operating system **120**, one or more applications **122**, and other program code and/or data components illustrated by other libraries and subsystems **124**. In an aspect, some or all software components may be stored in locations accessible to processor **104** in a shared memory address space shared by the software components. The software components accessed via the shared memory address space are stored in a shared processor memory defined by the shared memory address space. In another aspect, a first software component may be stored in one or more locations accessed by processor **104** in a first address space and a second software component may be stored in one or more locations accessed by processor **104** in a second address space. The first software component is stored in a first processor memory defined by the first address

space and the second software component is stored in a second processor memory defined by the second address space.

[0042] Software components typically include instructions executed by processor **104** in a computing context referred to as a "process". A process may include one or more "threads". A "thread" includes a sequence of instructions executed by processor **104** in a computing sub-context of a process. The terms "thread" and "process" may be used interchangeably herein when a process includes only one thread.

[0043] Execution environment **102** may receive user-provided information via one or more input devices illustrated by an input device **128**. Input device **128** provides input information to other components in execution environment **102** via input device adapter **110**. Execution environment **102** may include an input device adapter for a keyboard, a touch screen, a microphone, a joystick, a television receiver, a video camera, a still camera, a document scanner, a fax, a phone, a modem, a network interface adapter, and/or a pointing device, to name a few exemplary input devices.

[0044] Input device **128** included in execution environment **102** may be included in device **100** as FIG. **1** illustrates or may be external (not shown) to device **100**. Execution environment **102** may include one or more internal and/or external input devices. External input devices may be connected to device **100** via corresponding communication interfaces such as a serial port, a parallel port, and/or a universal serial bus (USB) port. Input device adapter **110** receives input and provides a representation to bus **116** to be received by processor **104**, physical processor memory **106**, and/or other components included in execution environment **102**.

[0045] An output device **130** in FIG. **1** exemplifies one or more output devices that may be included in and/or that may be external to and operatively coupled to device **100**. For example, output device **130** is illustrated connected to bus **116** via output device adapter **112**. Output device **130** may be a display device. Exemplary display devices include liquid crystal displays (LCDs), light emitting diode (LED) displays, and projectors. Output device **130** presents output of execution environment **102** to one or more users. In some embodiments, an input device may also include an output device. Examples include a phone, a joystick, and/or a touch screen. In addition to various types of display devices, exemplary output devices include printers, speakers, tactile

output devices such as motion-producing devices, and other output devices producing sensory information detectable by a user. Sensory information detected by a user is referred herein to as "sensory input" with respect to the user.

[0046] A device included in and/or otherwise providing an execution environment may operate in a networked environment communicating with one or more devices via one or more network interface components. The terms "communication interface component" and "network interface component" are used interchangeably herein. FIG. **1** illustrates network interface adapter (NIA) **114** as a network interface component included in execution environment **102** to operatively couple device **100** to a network. A network interface component includes a network interface hardware (NIH) component and optionally a network interface software (NIS) component.

[0047] Exemplary network interface components include network interface controller components, network interface cards, network interface adapters, and line cards. A node may include one or more network interface components to interoperate with a wired network and/or a wireless network. Exemplary wireless networks include a BLUETOOTH network, a wireless 802.11 network, and/or a wireless telephony network (e.g., a cellular, PCS, CDMA, and/or GSM network). Exemplary network interface components for wired networks include Ethernet adapters, Token-ring adapters, FDDI adapters, asynchronous transfer mode (ATM) adapters, and modems of various types. Exemplary wired and/or wireless networks include various types of LANs, WANs, and/or personal area networks (PANs). Exemplary networks also include intranets and internets such as the Internet.

[0048] The terms "network node" and "node" in this document both refer to a device having a network interface component for operatively coupling the device to a network. Further, the terms "device" and "node" used herein may refer to one or more devices and nodes, respectively, providing and/or otherwise included in an execution environment unless clearly indicated otherwise.

[0049] The user-detectable outputs of a user interface are generically referred to herein as "user interface elements" or abbreviated as "UI elements". More specifically, visual outputs of a user interface are referred to herein as "visual interface elements". A visual interface element may be a visual output of a graphical user interface (GUI).

Exemplary visual interface elements include windows, textboxes, sliders, list boxes, drop-down lists, spinners, various types of menus, toolbars, ribbons, combo boxes, tree views, grid views, navigation tabs, scrollbars, labels, tooltips, text in various fonts, balloons, dialog boxes, and various types of button controls including check boxes and radio buttons. An application interface may include one or more of the elements listed. Those skilled in the art will understand that this list is not exhaustive. The terms “visual representation”, “visual output”, and “visual interface element” are used interchangeably in this document. Other types of UI elements include audio outputs referred to as ”audio interface elements”, tactile outputs referred to as “tactile interface elements”, and the like.

[0050] A visual output may be presented in a two-dimensional presentation where a location may be defined in a two-dimensional space having a vertical dimension and a horizontal dimension. A location in a horizontal dimension may be referenced according to an X-axis and a location in a vertical dimension may be referenced according to a Y-axis. In another aspect, a visual output may be presented in a three-dimensional presentation where a location may be defined in a three-dimensional space having a depth dimension in addition to a vertical dimension and a horizontal dimension. A location in a depth dimension may be identified according to a Z-axis. A visual output in a two-dimensional presentation may be presented as if a depth dimension existed allowing the visual output to overlie and/or underlie some or all of another visual output.

[0051] An order of visual outputs in a depth dimension is herein referred to as a "Z-order". The term “Z-value” as used herein refers to a location in a Z-order. A Z-order specifies the front-to-back and/or back-to-front ordering of visual outputs in a presentation space with respect to a Z-axis. In one aspect, a visual output with a higher Z-value than another visual output may be defined to be on top of or closer to the front than the other visual output. In another aspect, a visual output with a lower Z-value than another visual output may be defined to be on top of or closer to the front than the other visual output.

[0052] A “user interface (UI) element handler” component, as the term is used herein, includes a component of configured to send information representing a program entity for presenting a user-detectable representation of the program entity by an output

device, such as a display. A “program entity” is an object included in and/or otherwise processed by an application or executable. The user-detectable representation is presented based on the sent information. Information that represents a program entity for presenting a user detectable representation of the program entity by an output device is referred to herein as “presentation information”. Presentation information may include and/or may otherwise identify data in one or more formats. Exemplary formats include image formats such as raw pixel data, JPEG, video formats such as MP4, markup language data such as hypertext markup language (HTML) and other XML-based markup, a bit map, and/or instructions such as those defined by various script languages, byte code, and/or machine code. For example, a web page received by a user agent from a remote application provider may include HTML, ECMAScript, and/or byte code for presenting one or more UI elements included in a user interface of the remote application. Components configured to send information representing one or more program entities for presenting particular types of output by particular types of output devices include visual interface element handler components, audio interface element handler components, tactile interface element handler components, and the like.

[0053] A representation of a program entity may be stored and/or otherwise maintained in a presentation space. As used in this document, the term “presentation space” refers to a storage region allocated and/or otherwise provided for storing presentation information, which may include audio, visual, tactile, and/or other sensory data for presentation by and/or on an output device. For example, a buffer for storing an image and/or text string may be a presentation space as sensory information for a user. A presentation space may be physically and/or logically contiguous or non-contiguous. A presentation space may have a virtual as well as a physical representation. A presentation space may include a storage location in a processor memory, secondary storage, a memory of an output adapter device, and/or a storage medium of an output device. A screen of a display, for example, is a presentation space.

[0054] As used herein, the terms "program” and "executable” refer to any data representation that may be translated into a set of machine code instructions and may optionally include associated program data. The terms are used interchangeably herein.

Program representations other than machine code include object code, byte code, and source code. Object code includes a set of instructions and/or data elements that either are prepared for linking prior to loading or are loaded into an execution environment. When in an execution environment, object code may include references resolved by a linker and/or may include one or more unresolved references. The context in which this term is used will make clear the state of the object code when it is relevant. This definition can include machine code and virtual machine code, such as Java™ byte code. As used herein, the terms “application”, and “service” may be realized in one or more executables and/or in one or more hardware components. The terms are used interchangeably herein.

[0055] As used herein, an “addressable entity” is program or a portion thereof, specifiable in programming language in source code. An addressable entity is addressable in a program component translated for a compatible execution environment from the source code. Examples of addressable entities include variables, constants, functions, subroutines, procedures, modules, methods, classes, objects, code blocks, and labeled instructions. A code block includes one or more instructions in a given scope specified in a programming language. An addressable entity may include a value. In some places in this document “addressable entity” refers to a value of an addressable entity. In these cases, the context will clearly indicate that the value is being referenced.

[0056] Addressable entities may be written in and/or translated to a number of different programming languages and/or representation languages, respectively. An addressable entity may be specified in and/or translated into source code, object code, machine code, byte code, and/or any intermediate languages for processing by an interpreter, compiler, linker, loader, and/or other analogous tool.

[0057] As used herein, the term “network protocol” refers to a formal set of rules, conventions and data structures that governs how computers and other network devices exchange information over a network. The rules, conventions, and data structures are said to be specified or defined in a specification and/or schema.

[0058] The term "network path" as used herein refers to a sequence of nodes in a network that are communicatively coupled for transmitting data in one or more data units of a specified network protocol between a pair of nodes in the network.

[0059] A "data unit", as the term is used herein, is a data entity specified according to a network protocol for transmitting data contained in and/or otherwise identified by the data entity. A data unit is transmitted between a pair of nodes in a network path to send the data in and/or otherwise identified by the data unit from a source node to a destination node that includes a protocol endpoint of the network protocol. A network protocol explicitly and/or implicitly specifies and/or otherwise identifies a schema that defines one or more of a format for a valid data unit and a vocabulary for content of a valid data unit. One example of a data unit is an Internet Protocol (IP) packet. The Internet Protocol defines a format for an IP packet that defines a header for identifying a destination address that identifies a destination node and a payload portion for including a representation of data to be delivered to the identified destination node. The terms "data unit", "data packet", and "packet" are used interchangeably herein. One or more data units of a first network protocol may transmit a message of second network protocol. For example, one or more data units of the IP protocol may include a TCP message. In another example, one or more TCP data units may transmit an HTTP message.

[0060] How data is packaged in one more data units for a network protocol may vary as the data traverses a network path from a source node to a destination node. Data may be transmitted in a single data unit between two consecutive nodes in a network path. Additionally, data may be exchanged between a pair of consecutive nodes in several data units each including a portion of the data. Data received in a single data unit by a node in a network path may be split into portions included in several respective data units for transmitting to a next node in the network path. Portions of data received in several data units may be combined into a single data unit for transmitting by a node in a network path. For purposes of describing the subject matter, a data unit in which data is received by a node is referred to as a different data unit than a data unit in which the data is forwarded by the node.

[0061] A “protocol address”, as the term is used herein, for a network protocol is an identifier of a protocol endpoint that may be represented in a data unit of the protocol. For example 192.168.1.1 is an IP protocol address represented in a human readable format that may be included in an address portion of an IP header to identify a source and/or a destination IP protocol endpoint. A protocol address differs from a symbolic identifier, defined below, in that a symbolic identifier, with respect to a network protocol, maps to a protocol address. Thus, “www. mynode.com” may be a symbolic identifier for a node in a network when mapped to the protocol address 192.168.1.1. An identifier may be both a symbolic identifier and a protocol address depending on its role with respect to its use for a particular network protocol.

[0062] Since a protocol endpoint is included in a node and is accessible via a network via a network interface, a protocol address identifies a node and identifies a network interface of the node. A network interface may include one or more NICs operatively coupled to a network.

[0063] A node in a pair of nodes in a network path at one end of the sequence of nodes in the network path and/or the other end is referred to herein as a “path end node”. Note that a node may have two NICs with one NIC at each end of a network path. A network path may be included as a portion of another network path that communicatively couples a same pair of nodes. Data may be transmitted via the sequence of nodes in a network path between path end nodes communicatively coupled via the network path. Data may be transmitted in one or both directions depending on an ordering of the nodes in the sequence.

[0064] The term “hop” as used herein refers to a pair of consecutive nodes in a network path for transmitting, via a network protocol, data sent from a source node to a destination node. A “hop path” is thus a sequence of hops in a network that respectively include a sequence of pairs of consecutive nodes included in transmitting data from a first path end node of the network path to a second path end node of the network path.

[0065] The term “path-based protocol address” as used herein refers to a protocol address for a network protocol that includes a plurality of path segment identifiers that identify portions of a specific network path identified by the path-based protocol address. A “node-based protocol address” is a path-based protocol address that

includes a plurality of node identifiers that identify a sequence of nodes in a network path. A "network-interface-based protocol address" is a path-based protocol address that includes a plurality of network interface identifiers that identify a sequence of network interfaces in a network path. A "NIC-based protocol address" is a type of network-interface-based protocol address that includes a plurality of identifiers that identify a sequence of network interface components. A "hop-based protocol address" is a type path-based protocol address since a hop is a type of network path.

[0066] Given the above definitions, note that the terms "network path" and "hop" may be defined in terms of network interfaces. A "network path" and a "hop path" include a sequence of network interfaces in a network that are included in transmitting data between a pair of path end nodes in the network. A "hop" refers to at least part of a network path that includes a pair of consecutive network interfaces in a sequence of network interfaces in a network path. A "network path" is thus a sequence of hops in a network that respectively include a sequence of pairs of consecutive network interfaces included in transmitting data from a first path end node of the network path to a second path end node of the network path.

[0067] The term "network topology" as used herein refers to a representation of protocol endpoints and/or nodes in a network, and representations of hops representing communicative couplings between and/or among the protocol endpoints and/or nodes in the network. A network may have different network topologies with respect to different network protocols. A network topology may represent physical communicative couplings between nodes in the network. A network topology may represent logical couplings between protocol endpoints and/or nodes of a particular network protocol or a particular type of network protocol.

[0068] The domain name system (DNS) of the Internet operates based on an application layer protocol defined by the DNS. The nodes in the DNS are communicatively coupled via the DNS protocol and may be represented by a logical network topology. A DNS system includes nodes connected via the DNS protocol and thus has a network topology defined by nodes that include protocol endpoints of the DNS protocol. In still another example, a token-ring network has a circular topology at the link layer, but may have a star topology at the physical layer.

[0069] As used herein, an "entity-specific address space" refers to an address space defined for a specific entity where the addresses in the address space operate as identifiers in the context of the entity. An address from an entity-specific address space is referred to herein as an "entity-specific address". An address is "entity-specific" in that what it identifies is based on the entity to which it is specific. Another address having the same form and content may identify a different entity when in an address space specific to another entity. Addresses in an entity-specific address space operate as identifiers in the context of an entity to which they are "specific" as defined by the specific association of the address space and the entity. Without knowledge of the entity to which an entity-specific address space is specific, what an address in the entity-specific address space identifies is indeterminate. The terms "entity-specific address" and "entity-specific identifier" are used interchangeably herein. An entity-specific address may identify an entity included in the entity to which the address is specific or may identify an entity external to the entity to which the address is specific. The fact that an address is entity-specific does not define a scope for the address.

[0070] A portion of a network is a type of entity. A type of entity-specific address space described herein is a scope-specific address space. As used herein, a "scope-specific address space", specific to a particular region of a network, is an address space defined for the particular network region, where an address in the scope-specific protocol address operates as identifier, according to a network protocol, of a protocol endpoint in a node outside of the particular region when processed in the context of a node in the particular region. The region is indicated by the span of an indicated scope. The terms "region" and "zone" are used interchangeably herein. An address from a scope-specific address space is referred to herein as a "scope-specific protocol address". An address is "scope-specific" in that what protocol endpoint it identifies depends on the region to which it is specific. Another address having the exact same form and content may identify a different protocol endpoint when in an address space that is specific to another region. A protocol address in a scope-specific address space serves as an identifier in the context of a node in a region to which the scope-specific address space is "specific" as defined by an association of the address space and the region indicated by the scope. Without knowledge of the particular region to which a scope-specific

address space is specific, what a scope-specific protocol address in the scope-specific address space identifies is indeterminate. The terms “scope-specific protocol address” and “scope-specific protocol identifier” are used interchangeably herein. Types of scope-specific address spaces indicating exemplary spans include site-specific, LAN-specific, subnet-specific, city-specific, business-specific, and node-specific.

[0071] For a network protocol, an address in a scope-specific address space serves as an identifier of a protocol endpoint in a node. Data may be received via the protocol endpoint from a network via one or more network interfaces that operatively couple the node to the network. Data may be sent via the protocol endpoint for transmitting over the network via the one or more network interfaces in the node. Since a protocol endpoint of a network protocol is included in a node and is accessible via a network via a network interface, a protocol address identifying the protocol endpoint also identifies the node and identifies a network interface of the node.

[0072] As used herein, a “node-specific address space” is a scope-specific address space defined for a specific node in a network, where the addresses in the node-specific address space operate as identifiers of nodes and/or network interfaces in the network when processed in the context of the specific node. An address from a node-specific address space is referred to herein as a “node-specific address”. An address is “node-specific” in that what it identifies depends on the node to which is defined as specific. Another address having the exact same form and content may identify a different node when in an address space specific to another node. Addresses in a node-specific address space operate as identifiers in the context of a node to which they are “specific” as defined by the specific association of the address space and the node. Without knowledge of the node to which a node-specific address space is specific, addresses in the node-specific address space are indeterminate. The terms “node-specific address” and “node-specific identifier” are used interchangeably herein. A node-specific address space is a type of scope-specific address space.

[0073] The term “node” is defined above. Note that an identifier of a network interface in a network also identifies a node that includes the network interface. Thus, a network interface-specific address is also a node-specific address. Network interfaces in a node may have their own respective network interface-specific address spaces that are also

node-specific. The network interface-specific address spaces may be combined to form a node-specific address space and/or may be managed as separate address spaces. The adjectives “node-specific” and “network interface-specific” may be used interchangeably.

[0074] A scope-specific identifier differs from a scoped address as described in "Request for Comments" (RFC) document RFC 4007 by S. Deering, et al, titled "IPv6 Scoped Address Architecture”, published by the IETF in December, 2006 and further described in Application No. 11/962,285, by the present inventor, filed on 2007/12/21, entitled “Methods and Systems for Sending Information to a Zone Included in an Internet Network”. A scoped address space is shared by nodes in a given scope. While a link-local scoped address is specific to a particular node, a link-local scoped address simply identifies a network interface component local to the particular node. A loop-back internet address is specific to a node as well. Neither link-local scoped addresses nor loop-back addresses identify one node to another. As such, neither serves as a node-specific identifier as defined above.

[0075] A “scoped address” is described by RFC 3513 and RFC 4007 as an identifier that, in a particular region of a network, serves as a protocol address of a network interface and/or a node in the particular region. The extent of the particular region is referred to as the scope of the region and thus the scope within which the identifier serves as a protocol address. A particular region included within a scope as indicated by its span. A scoped address is a valid protocol address only within a particular region as indicated by the address’s indicated scope. Examples of scope indicators include node-scope where identifiers are valid only to a single node in the indicated span, LAN-scope where identifiers are valid for nodes in the span of a particular LAN, and subnet-scope where identifiers are valid only for nodes in a particular subnet. RFC 3513 currently defines support for link-local scope, site-local scope, a global scope. A data unit transmitted with a scoped address should not be delivered to node that does not have a network interface in the span indicated by the scope.

[0076] “Path information” is any information that identifies a network path and/or hop path for data transmitted via one or more specified network protocols. Path information may be identified by identifying network interfaces, NICs, nodes, and/or hops included

in a network path. “Address information” is any information that identifies a protocol address that, for a network protocol, identifies a protocol endpoint. Address information may identify a unicast protocol address for a network protocol. In identifying a protocol endpoint, a protocol address identifies a node and a network interface. Those skilled in the art will understand upon reading the descriptions herein that the subject matter disclosed herein is not restricted to the network protocols described and/or their corresponding OSI layers. For ease of illustration, the subject matter is described in terms of protocols that correspond to OSI layer three, also referred to as network layer protocols, in general. Particular descriptions are based on versions of the Internet Protocol (IP). Address information may identify one or more protocol addresses. Exemplary protocol addresses include IP addresses, IPX addresses, DECNet addresses, VINES Internet Protocol addresses, and Datagram Delivery Protocol (DDP) addresses.

[0077] The term “path-based address” is defined above. A “node-based address” is a path-based address where some or all of the address includes node identifiers that identify a sequence of nodes in a network path. A “network-interface-based address” is a path-based address where some or all of the address identifies includes identifiers of a network interfaces in sequence in a network path. A “NIC-based address” is a type of network-interface-based address that identifies a sequence of network interface components. A “hop-based address” is a path-based address where some or all of the address identifies one or more hops in a network path. The protocol address types defined are not mutually exclusive.

[0078] The term “metric space”, as used herein, refers to a set, as defined in mathematics, where a distance between elements of the set is defined according to a metric. Metric spaces defined in Euclidean geometry are well-known examples. Those skilled in the art of metric spaces, such as Euclidian spaces, will appreciate that a one-to-one mapping may be determined and/or otherwise identified for mapping addresses from a first coordinate space having a first origin for a metric space to addresses from a second coordinate space having a second origin in the metric space. Given a mapping rule between a first scope-specific address space and a second scope-specific address space and a mapping between the second scope-specific address space and a third

scope-specific address space based on a third coordinate space identifying a third origin in the metric space, a mapping from the first coordinate space to the third coordinate space may be determined. A mapping between coordinate spaces for a metric space may be included a coordinate shift and/or a rotation, for example. The mapping may be pre-specified and accessible to the nodes in one or both address spaces. Mapping between locations in a number of different metric spaces is well-known in mathematics. For example, a top half of the surface of sphere may be mapped to a plane. Some will further appreciate that some metric spaces may be mapped to other metric spaces. Some of these mappings are one-to-one and onto.

[0079] An "interaction", as the term is used herein, refers to any activity including a user and an object where the object is a source of sensory data detected by the user. In an interaction the user directs attention to the object. An interaction may also include the object as a target of input from the user. The input from the user may be provided intentionally or unintentionally by the user. For example, a rock being held in the hand of a user is a target of input, both tactile and energy input, from the user. A portable electronic device is a type of object. In another example, a user looking at a portable electronic device is receiving sensory data from the portable electronic device whether the device is presenting an output via an output device or not. The user manipulating an input component of the portable electronic device exemplifies the device, as an input target, receiving input from the user. Note that the user in providing input is detecting sensory information from the portable electronic device provided that the user directs sufficient attention to be aware of the sensory information and provided that no disabilities prevent the user from processing the sensory information. An interaction may include an input from the user that is detected and/or otherwise sensed by the device. An interaction may include sensory information that is detected by a user included in the interaction that is presented by an output device included in the interaction.

[0080] As used herein "interaction information" refers to any information that identifies an interaction and/or otherwise provides data about an interaction between a user and an object, such as a portable electronic device. Exemplary interaction information may identify a user input for the object, a user-detectable output presented by an output

device of the object, a user-detectable attribute of the object, an operation performed by the object in response to a user, an operation performed by the object to present and/or otherwise produce a user-detectable output, and/or a measure of interaction. The term "operational component" of a device, as used herein, refers to a component included in performing an operation by the device.

[0081] Interaction information for one object may include and/or otherwise identify interaction information for another object. For example, a motion detector may detect user's head turn in the direction of a display of a portable electronic device. Interaction information identifying the user's head is facing the display may be received and/or used as interaction information for the portable electronic device indicating the user is receiving visual input from the display. The interaction information may serve to indicate a lack of user interaction with one or more other objects in directions from the user different than the detected direction, such as a person approaching the user from behind the user. Thus the interaction information may serve as interaction information for one or more different objects.

[0082] The term "attention information" as used herein refers to information that identifies an attention output and/or that includes an indication to present an attention output. Attention information may identify and/or may include presentation information that includes a representation of an attention output, in one aspect. In another aspect, attention output may include a request and/or one or more instructions for processing by a processor to present an attention output. The aspects described serve merely as examples based on the definition of attention information, and do not provide an exhaustive list of suitable forms and content of attention information.

[0083] As used herein the term "attention criterion" refers to a criterion that when met is defined as indicating that interaction between a user and an object is or maybe inadequate at a particular time and/or during a particular time period. In other words, the user is not directing adequate attention to the object.

[0084] As used herein, the term "tag" refers to a character string, which may include one or more words, which may be associated with a resource to create an association between the resource and another resource also tagged with the tag. Tags are often used in sharing media, social bookmarking, social news and blog entries to help users

search for associated content. In some contexts, the term “tagging” as used herein refers to the process of associating a tag with a resource that can be tagged. As used herein, the term “tagged association” refers to an association that identifies a tag and a resource that is tagged with the tag. A resource is said to be “tagged” with a tag when a tagging process, also referred to as “tagging”, has successfully created a tagged association. In other contexts, a “tagging” refers to a tagged association.

[0085] The term “matched tag”, as used herein, refers to an association that identifies a tag and a tagging criterion, where the tagging criterion is based on data other than the tag. The tagging criterion may be based additionally on the tag. The term “tagging association”, as used herein, refers to a matched tag represented in a memory, either persistent and/or volatile. For example, a tagging association may be realized as a record that includes and/or identifies a tag and a tagging criterion. The terms “matched tag” and “tagging association” are used interchangeably. When the term “matched tag” refers to something other than its tagging association, it will be made clear. For example, a reference to matched tag presented in a user interface clearly identifies a UI element that identifies and/or otherwise represents and/or otherwise identifies a matched tag either realized or to be realized in a tagging association. A tagged association, defined above, may identify a matched tag and a resource that is tagged with the tag identified by the matched tag. Such an association is also referred to herein as a “matched tag association”. A resource may be tagged with a matched tag when a tagging criterion identified by the matched tag is met for the resource.

[0086] As user herein, the term “vocabulary” refers to a set of valid values that can be assigned to and/or included in a data element. With respect to a tag, a vocabulary defines valid tags. A vocabulary may be specified by one or more rules and/or by identifying one or more valid values directly and/or indirectly.

[0087] As used herein, the term” dictionary” refers a vocabulary wherein one or more terms in the vocabulary is assigned a definition. A “definition” as the term is used herein refers to information that identifies semantic information about a data value, such as a t tag. Semantic information may include one or more of a textual description of a meaning of the term, an audio description, a visual description, and information identifying a use or context for the term. A use and/or context may be identified, for example, by

identifying a part of speech, a dialect or language, an antonym, a synonym, and/or an example usage of the defined data value.

[0088] As used herein, any reference to an entity "in" an association is equivalent to describing the object as "identified" by the association, unless explicitly indicated otherwise.

[0089] As used herein, the term "communication" refers to information including a message sent and/or for sending via a network between communicants. A message may include text data, audio data, and/or image data. The term "communicant" as used herein refers to a user included in a communication as a sender and/or a receiver of the information. A communicant is represented by a "communications agent" configured to operate in an execution environment to send data to and/or receive data from another communications agent, on behalf of the represented communicant, according to a communications protocol via network. A communications protocol defines and/or otherwise identifies an address space including communications addresses for delivering data sent in a communication from one communications agent to another. A communications protocol is a type of network protocol.

[0090] The term "communicant alias" as used herein refers to an identifier of a communicant in a communication where the communicant alias is not a communications address included in an address space of a communications protocol for sending and/or receiving data in the communication.

[0091] The term "attachment" as used herein refers to a portion of a communication that includes data from one communicant to another other than data in the message portion. A resource sent as an attachment is data that is typically not presented "inline" or in a message included in a message portion of a communication. Email attachments are perhaps the most widely known attachments included in communications. An email attachment is a file or other resource sent along with an email in a portion of the email separate from a message portion. A communication may include one or more resources as one or more attachments.

[0092] The terms "contactor" and "contactee" identify roles in a communication. A "contactor" provides information for identifying a "contactee" in a communication. A contactee may be included in a communication by a contactor and/or the contactor's

communications agent. A contactor and/or a contactee are roles of a user or communicant of a communications agent and may played by a living being, a node, a component, and/or an application. Both a contactor and a contactee are communicants in a communication.

[0093] As used here, the term "tagging- attribute association" refers to an association that identifies one or more tagging criterion and a resource attribute that the one or more tagging criterion are each based on. That is, whether a tagging criterion identified in a tagging-attribute association is met for a specified resource depends on an attribute of the resource identified by the tagging-attribute association. The term attribute, as used herein, includes the resource, as a resource is considered to be an attribute of itself and/or otherwise to include at least some of its attributes.

[0094] As used herein, the term "resource scope" refers a bounded space in which a resource may be identified. The space may be any space in which a resource may be locate and/or otherwise identified. A bounded may be may include one or more of an identifier space such as file system identifier space and/or a URI address space, a storage space, a geospatial space, a demographic space, and a data type – to name a few examples. space, defines a set of resources from which resources may be identified. For example, the files within a specified folder and its subfolders in a file system define and/or otherwise specify a resource scope. All user accounts that have an administrator security role define another resource scope. The present disclosure describes a resource scope with respect to resources tagged with a specified tag that makeup a set, referred to herein as a "tagged resource set". A tagged resource set may include a resource tagged with a tag identified by a matched tag.

[0095] In specifying a resource scope, one or more criterion may be specified. The terms "criterion" and "constraint" are used interchangeably herein, with respect to a resource scope. A criterion included in specifying a resource scope is referred to herein as a "scope criterion". A scope criterion may be expressed via any suitable manner, such as a file name, a wildcard expression, a regular expression, a data store location, a data base query, geospatial coordinates, a time, a date, a duration, a relationship to a user, and/or an address – too name a few examples. With respect to a tag, a scope criterion included in defining a resource scope differs from a tagging criterion for the tag

in that the scope criterion included in defining a resource scope need not be met in order for a resource to be tagged with the tag. Further a resource scope may be used for purposes other than processing related to a tagged resource.

[0096] One or more scope criterion that together specify and/or otherwise define a resource scope is/are referred to herein as a "scope descriptor". When all the criteria in a scope descriptor are met for a resource, the resource is in the resource scope defined by the scope descriptor. The one or more separate criterion in a scope descriptor may be evaluated in any order suitable or desirable in any given embodiment. In an aspect described below, one scope criterion, that may be included in a scope descriptor, may be specified to determine and/or otherwise identify when the scope descriptor is "active". A scope criterion that indicates whether a scope descriptor is active is referred to herein as a "scope condition". In a further aspect a scope descriptor may include no other criterion indicating the resource scope includes all resources accessible to an application, subsystem, execution environment, and/or group of any of the preceding that are processing and/or otherwise making use of the scope descriptor. Those skilled in the art will understand that whether a scope criterion in a scope descriptor is called a scope condition or not depends on the role played by the scope criterion. The subject matter may be described without referring to a scope criterion in the role of a scope condition without limiting the scope of subject matter.

[0097] Although, those skilled in the art will know that a scope criterion may be defined as "met" when it is true or may be defined as "met" when it is not true; for purposes of the present disclosure, A scope criterion is said to be "met" for a resource when it indicates the resource is in the resource scope defined by the scope descriptor, and is to be "not met" otherwise. A scope criterion in the role of a scope condition in a scope descriptor is said to be "met" when it indicates the scope descriptor is active, as is said to be "not met" otherwise. In an aspect, a scope descriptor may not include a scope condition or may include a scope condition that is always met. Such a scope descriptor is active by definition. Whether a criterion, in a scope descriptor is met or is not met, may depend on an evaluation of the scope criterion that is based on an identifiable entity. For a scope condition, such an entity is referred to herein as a "condition entity". More generally, for a scope criterion, such an entity is referred to herein as a "scoping

entity". Exemplary entities include a user, a document, a measure of time, a location, and the like.

[0098] An association between a tag and scope descriptor is referred to herein as a "scoped tag". A scoped tag is also referred to herein as a match set. As defined above, a tagging criterion may be used to determine whether to tag a particular resource with a particular tag. The tagging criterion is associated with the tag by a matched tag. A scope descriptor identifies resource scope that includes a set of resources. Scope descriptors are necessarily associated with resources.

[0099] A "tagging context", as the term is used herein, identifies and/or otherwise includes a set of tagged associations or taggings. The taggings may identify one or more tags and one or more resources. A "tagging context specification", as the term is used herein, is data that defines and/or otherwise specifies a tagging context. A tagging context specification may be stored in a memory location. The memory location may be contiguous or not and may be included in one or more data storage media, which may be volatile or not. A tagging context specification may include and/or otherwise identify any suitable criterion, referred to in general as "context criterion", for determining whether a tagging is in a tagging context defined by the tagging context specification. For example, a tag contact specification may identify one or more scope descriptors, one or more tags some or all of which may be and/or may be identified by matched tags, and a context condition; all of which are context criterion by definition.

[0100] A "context condition", as the term is used herein, refers to a context criterion that when met indicates that a tagging context is active for one or more taggings. When a context condition is not met, it is said to be inactive or not active for one or more taggings. For example, a context condition in a tagging context specification may be met for a first user. Taggings in the tagging context are "active" for the first user. For a second user, the context condition may not be met. The taggings are not active for the second user. A context condition, or more generally a context criterion, may be based on any suitable data for any suitable entity. A tagging context may be active for a task and not for another and/or a tagging context may be active for one location and not for another, to name a couple of other examples. When a context condition indicates that

an associated tagging context is active for a tag, the context condition is herein said to be "met", otherwise the context condition is not met.

[0101] FIG. **3A** illustrates an arrangement of components in a system that operates in an execution environment, such as execution environment **102** in FIG. **1**. The arrangement of components in the system operates to perform the method illustrated in FIG. **2A**. The system illustrated includes a query handler component **302**, a identification request constructor (IRC) component **304**, and a result handler component **306**. A suitable execution environment includes a processor, such as processor **104**, to process an instruction in at least one of a query handler component, a identification request constructor component, and a result handler component. FIG. **3B** illustrates an arrangement of components in a system that operates to perform the method illustrated in FIG. **2B**. The system illustrated includes a content manager component **309**, a query handler component **302**, a query director component **312**, and a identification response builder (IRB) component **314**. A suitable execution environment includes a processor, such as processor **104**, to process an instruction in at least one of a content manager component, a query handler component, a query director component, and a identification response builder (IRB) component.

[0100] Some components, illustrated in the drawings are identified by numbers with an alphanumeric suffix. A component may be referred to generically in the singular or the plural by dropping a suffix of a portion thereof of the component's identifier. For example, window UI elements, such as a window UI element **602a** in FIG. **6A**, a window UI element **602b** in FIG. **6B**, and their adaptations and analogs; are referred to herein generically as a window UI element **602** or window UI elements **602** when describing more than one. Other components identified with an alphanumeric suffix may be referred to generically or as a group in a similar manner.

[0101] The arrangement components illustrated in FIG. **4** may be adapted to operate in a number of execution environments to perform the method illustrated in FIG. **2A** and/or the method illustrated in FIG. **2B**. FIG. **4** is a block diagram illustrating the components of FIG. **3A**, FIG. **3B**, and/or analogs of the components of FIG. **3A** and FIG. **3B**, respectively adapted to operate in an execution environment **401** that includes and/or otherwise is provided by one or more nodes. FIG. **1** illustrates key components of an

exemplary device that may at least partially provide and/or otherwise be included in an execution environment. The components illustrated in FIG. **4** may be included in or otherwise combined with the components of FIG. **1** to create a variety of arrangements of components according to the subject matter described herein.

[0102] As stated, the various adaptations of the arrangement in FIG. **3A** as well as the various adaptations of the arrangement in FIG. **3B** illustrated and described herein are not exhaustive. For example, those skilled in the art will see based on the description herein that arrangements of components for performing the method illustrated in FIG. **2A** and the method illustrated in FIG. **2B** may each be distributed across more than one node and/or execution environment. For example, such an arrangements may operate at least partially in a browser in a one node and at least partially in a server in another node interoperating via a network.

[0103] FIG. **5** illustrates a first node **502** and an execution environment **401** of second node **504** as exemplary devices that each may be included in and/or otherwise adapted for providing an instance, adaptation, and/or analog of execution environment **401** in FIG. **4**. As illustrated in FIG. **5**, first node **502** and an execution environment **401** of second node **504** are operatively coupled to a network **506** via respective network interface components enabling first node **502** and execution environment **401** of second node **504** to exchange data in a communication on behalf of communicants represented by the respective nodes.

[0104] FIG. **4** illustrates communications agent **403** including an adaptation of the arrangement of components in FIG. **3A** and an adaptation of the arrangement of components in FIG. **3B**. A first communications agent **403** may operate, in first node **502**, on behalf of a first communicant to communicate with a second communicant represented by a second communications agent **403** operating in execution environment **401** of second node **504**.

[0105] Execution environment **401** is illustrated in FIG. **4** hosting communications agent **403**. Components in FIG. **4** are referenced for illustrative purposes in describing communications agents operating in an execution environment of first node **502** and an execution environment of execution environment **401** of second node **504**. Exemplary communications agents include email clients, phone clients including Voice over

Internet Protocol (VoIP) clients, instant messaging clients, short message service (SMS) clients, multimedia message service (MMS clients), multi-media communications clients including video phone clients, and other data transfer agents.

[0106] Communications agents **403** in FIG. **4**, respectively operating in first node **502** and in execution environment **401** of second node **504** in FIG. **5** may interoperate via respective network stacks **405**. Communications agents **403** may communicate via one or more communications protocols. FIG. **4** illustrates communications protocol component **407** exemplifying a subsystem for exchanging data via network **506** according to one or more communications protocols, such as simple mail transfer protocol (SMTP), an instant messaging protocol, and/or a real-time voice and/or video protocol. A communication between communications agents **403** in first node **502** and execution environment **401** of second node **504** may include more than one type of data and may use one or more communications protocols in exchanging one or more types of data via network **506**.

[0107] Instances, adaptations, and/or analogs of communications agent **403** in FIG. **4**, in first node **502** and in execution environment **401** of second node **504**, respectively, may communicate via discrete messages, a request/reply protocol, a data streaming protocol, a session and/or connection-oriented protocol, a connectionless protocol, a real-time communications protocol, an asynchronous communication, a store and forward communications protocol, a reliable delivery communications protocol, a best-effort delivery communications protocol, and/or a secure protocol, to name a few communications options.

[0108] FIG. **4** illustrates communications agent **403** including a content manager component **409**. Content manager component **409** may interoperate with communications protocol layer component **407** and/or network stack **405** to receive data in one or more communications via network **506** with another communications agent in another node. Content manager component **409** is operatively coupled, via com-in component **411**, to communications protocol component **407** for receiving the data from the other node.

[0109] Data received in a communication may include one or more resources and/or content types. Exemplary content types include plain text, markup such as hypertext

markup language (HTML), audio data, image data, and/or executable data. Executable data may include script instruction(s), byte code, and/or machine code. In FIG. **4**, communications agent **403** includes one or more content handler components **413** to process data received according to its content type. A data type may be identified by a MIME type identifier. Exemplary content handler components **413** include a text/html content handler component for processing HTML representations; an application/xmpp-xml content handler component for processing extensible messaging and presence protocol (XMPP) streams including presence tuples, instant messages, and audio content handlers including and/or configured to retrieve suitable codices; one or more video content handler components for processing video representations of various types; and still image data content handler components for processing various image data representations.

[0110] Content handler component(s) **413** process received data representations and may provide transformed data from the representations to one or more user interface element handler components **415**. One or more user interface element handler components **415** are illustrated in a presentation controller component **417** in FIG. **4**. Presentation controller component **417** may manage visual, audio, and other types of output for its including application as well as receive and route detected user and other inputs to components and extensions of its including application, communications agent **409**. A user interface element handler component **415** may be adapted to operate at least partially in a content handler component **413** such as a text/html content handler component and/or a script content handler component. Additionally or alternatively, a user interface element handler component in an execution environment **401** may be received in a communication. For example, a communication, such as an email, may include an HTML content type portion and a script content type portion.

[0111] FIG. **6A-G** illustrates various communications agent windows **602** presentable in a presentation space of a display device, such as output device **130** in FIG. **1**. In FIG. **6a**, a communications agent window **602a** includes a contactor user interface (UI) element **604a** for presenting an identifier of a communicant in the role of a contactor in a communication represented by the communications agent window **602a**. A communications agent window **602a** also includes a contactee UI element **606** for

presenting one or more contactee identifier(s) identifying one or more communicants in the role of contactee(s) included in the communication. A presentation space **608** is provided by each communications agent window **602a** for presenting a user message UI element **610** for presenting a message addressed to one or more contactees identified in a contactee UI element **606a**. A presentation space **608a** may also be provided for presenting one or more UI controls for exchanging data in and/or otherwise managing a communication. A send UI element **612a** illustrates an exemplary UI element that may correspond to user input to send data in a communication to one or more identified contactees.

[0112] Data to send in a communication to a communications agent in execution environment **401** of second node **504** may be received by one or more content handler component(s) **413** operating in first node **502** to transform the data into one or more data representations suitable for transmitting in the communication and/or suitable for processing by the communications agent in execution environment **401** of second node **504**. The one or more data representations may be provided to content manager component **409** for sending in the communication to execution environment **401** of second node **504**. Content manager component **409** may package the one or more data representations in a message formatted according to a communications protocol of the communications agent. Communications protocol component **407** may send the data according to the specification(s) of the communications protocol. Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in the communication to a communications agent via a network.

[0113] Content manager component **409** operating in execution environment **401** included in and/or otherwise provided by first node **502** may provide the packaged, encoded, and/or transformed data to communications protocol component **407** via a com-out component **419**. Com-out component **419** as described above operatively couples communications agent **409** to communications protocol component **407** according to an interface provided by communications protocol component **407** for sending data in a communication according to a communications protocol.

Communications protocol component **407** may further package and/or otherwise transform the data to send via network stack **405** for delivery via network **506** to execution environment **401** of second node **504**.

[0114] As described above, a communications protocol may operate via one or more nodes in a network in a network path including a first node and an execution environment **401** of second node **504**. Exemplary path nodes include mail relay nodes, phone switch nodes, and proxy nodes such as instant messaging proxies for communicating through firewalls. As indicated, path node **508** illustrates such a node.

[0115] Sending data in a communication requires identifying a contactee. A contactee may be represented by a communications address in an address space of a communications protocol. In one aspect, information identifying a communications address may be received from a user/communicant of a communications agent in a user node. In FIG. **4**, presentation controller **417** and/or a UI element handler **415**, presenting and/or managing interaction with contactor UI element **604** in FIG. **6A** for first node **502**, may receive a contactor alias in response to a user input corresponding to UI element handler **604**. The user/communicant of first node **502** may enter a contactor alias, such as “Dad”, via a keyboard and/or select a predefined communicant alias presented in a selection UI control element via a UI element handler component **415**. The user input may be detected by input driver **421**. Corresponding input information may be routed to presentation controller **417** by GUI subsystem **423**. GUI subsystem **423** may send presentation information to a display device via a graphics subsystem **425**. Communications agent **403** may identify a communication address associated with the contactor alias, “Dad”. Other communications address(es), such as for one or more contactees, may be received similarly and/or in any suitable manner.

[0116] Data may be sent in a communication according to a form or type of the communication and/or other attribute of the communication such as a security attribute, the amount of data to be set, a priority setting, a task setting, and the like. Some forms of communication do not require a session and/or connection between a first node and an execution environment **401** of second node **504** in order to send data between the two nodes, while others do. An email and/or instant message may use a store and forward model of delivery.

[0117] Data may be sent in a communication in response to a communicant input. A contactor may provide an input corresponding to send UI element **612a** in FIG. **6A**. The input may be received by presentation controller component **417** and/or one or more UI element handlers **415** corresponding to send UI element **612a**. In response to detecting the input, presentation controller component **417** may provide data to be sent in the communication to one or more content handler components **413** according to the content type(s) of the data to be sent.

[0118] The one or more content handler components **413** may encode, format, and/or otherwise transform the data for sending in a message, such as an email message. The one or more content handler components **413** may provide data to be sent to content manager **409**, instructing content manager component **409** to send the data in the communication for delivery to a communications agent in execution environment **401** of second node **504**. Content manager component **409** interoperating with com-out component **419** may further format and/or transform the data for sending in the communication according to communications protocol, for example according to an email communications protocol, by communications protocol component **407**. Communications protocol component **407** may send the communication for delivery to the communications agent in execution environment **401** of second node **504** via network **506**.

[0119] Path node **508** may relay data sent in the communication between first node **502** and execution environment **401** of second node **504**. Path node **508** may determine a next node and/or a network interface in a network path communicatively coupling first node **502** and execution environment **401** of second node **504** for exchanging data in a communication between communications agents **403** in the first node **502** and in the execution environment **401** of second node **504**.

[0120] For session-oriented and/or connection-oriented communication a session and/or connection may be established if a session/connection has not already been established. Data may be sent for delivery to a communications agent identified based on a contactee communications address during session and/or connection setup. For example, for a voice communication a voice communication may be established via a session initiation protocol. Communications protocol component **407** may operate

according to the session initiation protocol specifications. Communications protocol component **407** operating in first node **502** may locate a communications agent by communicating with one or more nodes in network **506** according to the session initiation protocol. Communications protocol component **407** may locate a communications agent in execution environment **401** of second node **504**, based on a communications address for the contactee located based on the contactor alias.

[0121] Once a communication session is established, such as a voice session, data may be sent according to the session communications protocol, such as RTP. Data may be sent according to the session initiation protocol in the communication for managing the voice communication session and/or for exchanging text, image, and/or other data outside of the voice session. Path node **508** may be included in session and/or connection setup. Alternatively or additionally, path node **508** may be included in a network path in a session and/or connection.

[0122] With reference to FIG. **2A**, a block **202** illustrates that the method includes receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. Accordingly, a system for processing a data object identification request in a communication includes means for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. For example, the arrangement in FIG. **3A**, includes query handler component **302** that is operable for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. FIG. **4** illustrates query handler component **402** as an adaptation and/or analog of the query handler component **302** in FIG. **3A**. One or more query handler components **402** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object

information identifying a data object matching criterion. In FIG. **4**, query handler component **402** is illustrated as a component of communications agent **403**.

[0123] FIG. **7** includes dataflow message **702** that illustrates a data flow internal to execution environment **401** of first node **502** and/or a message received via network **506** by execution environment **401** of first node **502**. Dataflow message **702** illustrates a data object information dataflow to query handler component **402** operating in execution environment **401** of first node **502**. Data object information is identified to and/or otherwise received by query handler component **402**.

[0124] In various aspects, a query handler component **402** may receive data object information from a user via one or more UI element handler components **415**. The data object information may be received, in response to a user input detected by an input device of an execution environment **401** that includes the query handler component **402**. Alternatively or additionally, data object information may be received by a query handler component in a message received via a network. For example, a user may identify data object information to a browser operating in a client node. The browser may send the data object information via a network communicatively coupling the client node to a web service provider node. The web service provider node may include a query handler component that may operate to receive the data object information identified in the message from the browser. FIG. **6A** illustrates an exemplary user interface that may be presented by communications agent **403** operating in execution environment **401** of first node **502**. A UI element handler component **415** may operate to present a user interface element to allow a user to identify data object information. A data object information (DOI) UI element **616a** is illustrated in FIG. **6A** as an example. The same or different UI element handler component **415** may present various other UI elements included in DOI UI element **616a**. DOI UI element **616a** and one or more UI elements it includes may be presented by one or more corresponding UI element handler component(s) **415** that may operate based on a schema that defines valid data object information and/or data object identification requests for including in a communication via a communications agent **403**. The schema defines one or more rules and/or a vocabulary that defines whether data object information and/or a data object identification request is valid. Those skilled in the art will understand that

numerous languages and/or schemas for providing data object information and/or data object identification requests currently exist including relatively simple key word based queries to relatively more complicated regular expression languages and data base query languages, such as various versions of structured query language (SQL). New schemas are and will be created, and are, thus, considered within the scope of the subject matter described herein.

[0125] DOI UI element **616a** illustrates a user interface for a keyword based schema. Match textbox UI element **618a** illustrates a textbox allowing a user to enter a keyword expression. Various checkbox UI elements **620a** allow a user to define a scope of a data object identification request by identifying whether a resulting data object identification request is to be applied to folders, files, and/or content of files in a file system and/or other data store of a receiving execution environment. A communications agent **403** may support more than one data object identification request schema and/or corresponding data object information schema. Thus. a communications agent may provide a user interface to receive valid data object information for each respective schema.

[0126] Returning to FIG. **2A**, a block **204** illustrates that the method further includes sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. Accordingly, a system for processing a data object identification request in a communication includes means for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. For example, the arrangement in FIG. **3A**, includes identification request constructor component 304 that is operable for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the

data object matching criterion, wherein the first message is addressed to the second user. FIG. **4** illustrates identification request constructor component **404** as an adaptation and/or analog of identification request constructor component **304** in FIG. **3A**. One or more identification request constructor components **404** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. In FIG. **4**, identification request constructor component **404** is illustrated as a component of communications agent **403**.

[0127] FIG. **7** includes dataflow message **704** that illustrates a data flow internal to execution environment **401** of first node **502** and/or message received via network **506** by execution environment **401** of first node **502**. Dataflow message **704** illustrates a create data object identification request (CIReq) dataflow to an IRC component **404** operating in execution environment **401** of first node **502**. Data object information may be identified to and/or otherwise provided to IRC component **404**. A query handler component **402**, operating as described with respect to block **202** of FIG. **2A**, may identify and/or otherwise provide data object information to an IRC component **404**.

[0128] A query handler component may be a type of content handler component that operates to process data object information to create a data object identification request that conforms to a schema for creating and/or otherwise constructing a valid data object identification request. The IRC component **404** may construct and/or otherwise create a data object identification request based on a schema that defines and/or otherwise identifies a valid data object identification request and/or a valid data object identification response for a particular type of communication supported by a communications agent. The data object identification response is constructed and/or otherwise created, based on the data object information, by the IRC component **404**. IRC component **404** in execution environment **401** of first node **502** may provide the

data object identification request to content manager component **409** to include and/or otherwise identify the data object identification request in a communication with execution environment **401** of second node **504**, such as email and/or instant message. Content manager component **409** may interoperate with other types of content handler components **413** to create and/or otherwise construct a message that includes the valid data object identification request.

[0129] In an aspect, the IRC component **404** may interoperate with the content manager component **409** to create the valid data object identification request according to a specified schema for the message, so that the message is valid according to a communications protocol supported by a communications protocol component **407** that operates to send the message. As described above, the content manager component **409** may operate in an execution environment **401** of first node **502** along with an IRC component **404** to transform data object information into a data object identification request to include along with data for other parts of a communication into one or more representations suitable for transmitting in a communication to another node, such as execution environment **401** of second node **504**. Some or all of the representations transmitted are suitable for processing by the communications agent in execution environment **401** of second node **504**. The content manager component **409** in the execution environment **401** of first node **502** may package the one or more data representations including a representation of the data object identification request into a message formatted according to the communications protocol.

[0130] In FIG. **8A**, a portion of an email communication **800a** is illustrated formatted as a multipart/mixed content type including data object identification request portion **802a**. A data object identification request portion of a communication may be identified as a data object identification request by its location in the communication and/or by an identifier or markup element, such as a MIME type identifier. A data object identification request may be detected based on content included in the message and/or based on metadata such as content-type header **804a** identifying a MIME type identifier, such as “application/id-request”, which may be defined for representing one or more matching criteria in a data object identification request. The “application/id-request” MIME type

identifier is exemplary. Other MIME type identifiers exist and/or may be defined to identify a data object identification request in a communication.

[0131] An IRC **404** and/or a content manager component **409** may operate to construct a data object identification request in a communication based on XQuery, regular expression, and/or SQL content – to name a few examples. A content type identifier may be included in a position and/or location that identifies a data object identification request in a communication. The position or location may be absolute or relative. For example, a schema for a communication may define that a data object identification request in a communication is included in the communication at the end of the communication. There may be one or more data object identification requests at the end. In another aspect, a schema for a communication may specify that a portion of a communication following a particular type of message portion is a data object identification request. Other data object identification requests may follow. If no data object identification request is included, the data object identification request portion may include no content or may include an indicator that no data object identification request is included.

[0132] FIG. **8A** illustrates an "application/id-request" MIME type identifier that may be defined to identify a schema for an XML-based language for specifying id-request XML documents. FIG. **8A** illustrates id-request document **806a**. Id-request document **806a**, as illustrated, includes criterion tag elements **808a** corresponding to the form elements in the user interface illustrated in FIG. **6A**. A criterion tag element **808a** identifies a data object matching criterion name, such as "query" indicating that the criterion is a query expression or a portion thereof. A "type" attribute identifies a schema with the identifier "keyword" for the expression. The criterion tag element **808a** identifies a value for a query "Uncle Joe" in a match-expression attribute. Another criterion tag element **808a** specifies a scope query specified in the first criterion tag element. FIG. **8A** illustrates an "and" tag **810a** indicating that all the matching criteria must be met in identifying a data object. An "or" tag (not shown) may be defined by a schema for id-request documents. Other operator elements and operator precedence may be defined by the schema. Grouping elements for managing operator precedence, such as a parenthesis element, may be defined by the schema.

[0133] FIG. **7** includes dataflow message **706** that illustrates a first message, including a data object identification request, sent via network **506** by execution environment **401** of first node **502** to execution environment **401** of second node **504**.

[0134] A data object identification request generated by an IRC component **404**, along with a user message and any other data to include in a communication, may be provided and/or otherwise identified to content manager component **409** for sending in the communication. The content manager component **409** in execution environment **401** of first node **502** may package the data including a representation of the data object identification request into the message formatted according to the communications protocol. Com-out component **419** may provide the data object identification request, the user message, and any other data for sending in the communication in representations suitable for sending by communications protocol component **407** to the communications agent in execution environment **401** of second node **504**. Communications protocol component **407** operating in execution environment **401** of first node **502** may send the data as and/or in a communication, such as an email message, according to the specification(s) of the communications protocol. Communications protocol component **407** may further package and/or otherwise transform the data to send via network stack **405** for delivery via network **506** to execution environment **401** of second node **504**.

[0135] Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in a communication with the communications agent in execution environment **401** of second node **504**.

[0136] Returning to FIG. **2A**, a block **206** illustrates that the method yet further includes receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. Accordingly, a system for processing a data object identification request in a communication includes means for

receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. For example, the arrangement in FIG. **3A**, includes result handler component **306** that is operable for receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. FIG. **4** illustrate result handler components **406** as an adaptations and/or analogs of result handler component **306** in FIG. **3A**. One or more result handler components **406** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. In FIG. **4**, a result handler component **406** is illustrated as a component of application **403**.

[0137] FIG. **7** includes dataflow message **714** that illustrates a second message, including a data object identification response, received via network **506** by execution environment **401** of first node **502** from execution environment **401** of second node **504**.

[0138] In addition to packaging communication information including a data object identification request into a valid communication, content manager component **409** operating in execution environment **401** of first node **502** may interoperate with communications protocol layer component **407** and/or network stack **405** to receive data in one or more communications via network **506** with another communications agent in another node. A result handler component **406** may receive some or all of the

communication in receiving and/or otherwise identifying a data object identification response in the communication. Content manager component **409** in FIG. **4** operatively couples result handler component **406** with network **506**, via com-in component **411**, communications protocol component **407**, and/or network stack **405** to receive the data from the other node. Thus, execution environment **401** of first node **502** may receive a communication, via content manager component **409** and result handler component **406**, that includes a data object identification response that is a response to a data object identification request detected by execution environment **401** of second node **504** in a message received previously from the execution environment **401** of first node **502**.

[0139] Com-in component **411** may provide data received in a communication to content manager component **409**. Content handler component **409** may identify a data object identification response in the communication sent from the execution environment **401** of second node **504**. Content manager component **409** may interoperate with result handler component **406** in identifying the data object identification response. The data object identification response may be identified by content manager component **409** according to a schema for the communication. For example, a portion of an email message may include a MIME-type identifier that identifies a corresponding part of the email as a data object identification response and/or otherwise identifies the part for routing to a result handler component **406** in communications agent **403** in execution environment **401** of first node **502** for detecting the data object identification response.

[0140] Data sent in second message **714**, by execution environment **401** of second node **504**, may be received by com-in component **411** operating in an instance, adaptation, and/or analog of execution environment **401** including and/or provided by first node **502**. The received communication includes a message addressed to the a communicant represented by communications agent **403** in first node **502** and a data object identification response identifying a data object located and/or otherwise identified based on the data object identification request identified by first message **706**. Content manager component **409** may detect the user message, such as message portion **802b** in second message **714**. Content manager component **409** may detect a data object identification response, such as data object identification response portion

804b in second message **714**. Message portion **802b** and data object identification response portion **804b** may be provided to suitable content handler components **413** based on the content types of the message portion and the data object identification response portion detected by content manager component **409**. In FIG. **4**, data object identification response portion **804b** may be provided to result handler component **406** as a content handler that matches the MIME type field **806b** illustrated in FIG. **8A**.

[0141] In an aspect, a data object identification response that is a response to a data object identification request sent in a previous message may include information identifying one or more data objects located and/or otherwise identified by the execution environment **401** of second node **504** that processed the data object identification request. For example, a data object identification response may be sent that identifies a URI for each data object located and/or otherwise identified by a first node in response to processing a corresponding data object identification request.

[0142] Returning to FIG. **2A**, a block **208** illustrates that the method yet further includes sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. Accordingly, a system for processing a data object identification request in a communication includes means for sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. For example, the arrangement in FIG. **3A**, includes access request constructor component **313** that is operable for sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. FIG. **4** illustrates access request constructor component **408** as an adaptation and/or analog of access request constructor component **308** in FIG. **3A**. One or more access request constructor components **408** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. In FIG. **4**, a access request

constructor component **408** is illustrated as a component of result handler component **406**.

[0143] FIG. **7** includes dataflow message **716** that illustrates a present message that may be internal to execution environment **401** of first node **502** and/or may be a message sent to a remote device, such as a node including a browser, to send presentation information to present, via an output device, a representation of the data object identification response received in second message **714**. FIG. **7** also includes dataflow message **718** that illustrates a create access request message that may be a data flow internal to execution environment **401** of first node **502** and/or a message received via network **506** by execution environment **401** of first node **502**. In the dataflow message **718** an ARC component **408** operating in execution environment **401** of first node **502** may be invoked to operate based on result information. Result information may be identified to and/or otherwise by ARC component **408** as part of and/or in response to being invoked.

[0144] Result handler component **406** may be a type of content handler component **413** that operates to process data representations and may provide transformed data from the representations to one or more user interface element handler components **415**. Result content handler component **406** may operate to process data object identification responses according to a schema defining valid data object identification responses. Content manager component **409** may provide and/or otherwise identify various portions of a communication, such as a data object identification response, to one or more content handler components **413**. The data object identification response may be provided to result handler component **404**. In an aspect, result handler component may interoperate with the presentation controller component **417** in the execution environment **401** of first node **502** to present the data received in the data object identification response and/or other data received in the communication.

[0145] FIG. **6B** illustrates a response window UI element **602b** as an exemplary user interface that may be presented by communications agent **403** operating in execution environment **401** of first node **502** to present data received in a communication from execution environment **401** of second node **504**. FIG. **6B** illustrates a contactor UI element **604b**, a contactee UI element **606b**, a presentation space **608b** of the

response window UI element **602b**. Presentation space **608** includes a user message UI element **610b**, analogous to similar UI elements described with respect to FIG. **6A**. The various UI elements may be presented by one or more UI element handler components **415** in the communications agent **402** in execution environment **401** of first node **502**. A reply UI element **622b** and a reply all UI element **624b** illustrate UI elements in presentation space **608b** presented to allow a user of the execution environment **401** of first node **502** to respond and/or otherwise instruct execution environment **401** of first node **502** to perform an operation based on the message received in the communication. FIG. **6B** illustrates a data object identification requested query UI interface element **626b** where the presented term "None" may be defined to indicate no data object identification request was detected by the content manager component **409** in execution environment **401** of first node **502** in the communication received by execution environment **401** of first node **502** from second node **504** via network **506**.

[0146] Response window UI element **602b** in FIG. **6B** includes a data object identification response UI element **628b** for receiving user input to display a representation of one or more data objects located and/or otherwise identified by the execution environment **401** of second node **504** and identified in the data object identification response. One or more of the data objects is not included in the message received in the communication. In response to an input directed to data object identification response UI element **628b**, a corresponding UI element handler **415** may interoperate with result handler component **406** to present a located data resources dialog UI element **630b** that includes a presentation space **632b** for presenting a user detectable representation of some or all of the data objects identified in the received data object identification response from the execution environment **401** of second node **504**. In an aspect, one or more data objects located and/or otherwise identified by execution environment **401** of second node **504** may be included in a communication as attachments. In a further aspect, located and/or otherwise identified data objects not included in the message may be identified as described with respect to located and/or otherwise identified data objects dialog UI element **630b** and/or in any other suitable manner. In yet another aspect, files and/or other data entities identified in the data

object identification response may be illustrated in a manner similar to attachments rather than in a separate UI element as in FIG. **6B**, and accessed from first node **504** via network **506** as described below.

[0147] FIG. **6B** illustrates an access UI element **634b** and an access all UI element **636b** allowing a user to provide user input(s) to instruct communications agent **403** in execution environment **401** of first node **502** to request one or more data objects identified in the data object identification response.

[0148] FIG. **7** includes a dataflow message **720** that illustrates a third message that includes an access request to access a data object identified in the data object identification response received in second message **714**. FIG. **7** illustrates the access request is sent via network **506** by execution environment **401** of first node **502** to execution environment **401** of second node **504**.

[0149] In response to receiving second message **714**, execution environment **401** of first node **502** may present, in a manner described above, data received in the message to a user of execution environment **401** of first node **502** as described with respect to FIG. **6C**. Context menu **646c** provides UI elements to allow the user to instruct execution environment **401** of first node **502** to retrieve a data object identified in a data object identification response received from execution environment **401** of second node **504**. In response to a user input corresponding to a context menu item included in context menu **646c** a corresponding UI element handler component **415** may invoke ARC component **408** to construct an access request to retrieve one or more data objects identified in the data object identification response. FIG. **6G** illustrates another exemplary user interface that execution environment **401** of first node **502** may present to the user to allow the user to instruct execution environment **401** of first node **502** to construct and send a request. Auto-response window **602g** may be presented in response to a data object identification response received in a communication without a user message. Rather than a UI element for displaying a user message, a results pane UI element **670g** may be presented that identifies data objects located and/or otherwise identified by execution environment **401** of second node **504** that may be retrieved by execution environment **401** of first node **502**. In an aspect, execution environment **401** of second node **504** may process a data object identification request and automatically

send a data object identification response without involving the user of execution environment **401** of second node **504**. The data object identification response may be received without a user message as FIG. **6G** illustrates or may be received with a pre-written and/or automatically generated message. UI elements **672g-674g** may be presented by execution environment **401** of first node **502** to allow the user of execution environment **401** of first node **502** to request the retrieval of one or more data objects identified in the data object identification response.

[0150] ARC component **408** may be invoked to construct an access request to retrieve one or more data objects identified in the data object identification response. An access request may be based on a URL, such as an HTTP and/or FTP URL that identifies and locates a data object. Alternatively or additionally, an access request may be sent in a communication supported by a communications agent **403**, in a manner analogous to that illustrated with respect to FIG. 9A. See U. S. Patent Application No. 12/833,014 (published US 2012-0011207 A1) filed on 2010/07/09, entitled "Methods, Systems, and Program Products for Processing a Request for a Resource in a Communication" and Application No. 12/833,016 (published US 2012-0011444 A1) filed on 2010/07/09, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR REFERENCING AN ATTACHMENT IN A COMMUNICATION". As described above, content manager component **409** in execution environment **401** of first node **502** may package the one or more data representations including a representation of an access request into a message in a communication. The message is formatted according to a communications protocol supported by communications agent **403** in execution environment **401** of first node **502**. Communications protocol component **407** operating in execution environment **401** of first node **502** may send the message according to the specification(s) of the communications protocol. Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in a communication.

[0151] Content manager component **409** operating in execution environment **401** included in and/or otherwise provided by execution environment **401** of first node **502** may provide the packaged, encoded, and/or transformed data to communications

protocol component **407** via a com-out component **419**. Com-out component **419** as described above operatively couples communications agent **409** to communications protocol component **407** according to an interface provided by communications protocol component **407** for sending data in a communication according to a communications protocol. Communications protocol component **407** may further package and/or otherwise transform the data to send via network stack **405** for delivery via network **506** to execution environment **401** of second node **504**.

[0152] Returning to FIG. **2A**, a block **210** illustrates that the method yet further includes receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. Accordingly, a system for processing a data object identification request in a communication includes means for receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. For example, the arrangement in FIG. 3, includes content director component **310** that is operable for receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. FIG. **4** illustrates content director component **410** as an adaptation and/or analog of content director component **310** in FIG. 3. One or more content director components **410** operate in an execution environment **401**. The system for content director includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. In FIG. **4**, a content director component **410** is illustrated as a component of communications agent **403**.

[0153] FIG. **7** includes dataflow message **722** that illustrates a fourth message that includes an access response received via network **506** by execution environment **401** of first node **502** from execution environment **401** of second node **504**.

[0154] As described above, content manager component **409** operating in execution environment **401** of first node **502** may interoperate with communications protocol layer component **407** and/or network stack **405** to receive data in a communication, via network **506**, with another communications agent in another node. Content manager

component **409** is operatively coupled, via com-in component **411**, to communications protocol component **407** to receive the data from the other node. Thus, execution environment **401** of first node **502** may receive a message in a communication with first node **504** via a com-in component **411**. The message includes an access response to an access request. Content director component **410** may be included in content manager component **409** to detect one or more portions of a received message, such as a portion that is a response to an access request. Content director component **410** may invoke one or more content handler components **413** to process one or more data objects provided in the message according to their type.

[0155] In one aspect, a data object returned in a message in response to access request may be processed as an attachment and/or in a manner analogous to processing of an attachment for a particular communications type.

[0156] With reference to FIG. **2B**, a block **212** illustrates that the method includes receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. Accordingly, a system for processing a data object identification request in a communication includes means for receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. For example, the arrangement in FIG. **3B**, includes content manager component **309** that is operable for receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. FIG. **4** illustrates content manager component **409** as an adaptation and/or analog of content manager component **309** in FIG. **3B**. One or more content manager components **409** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, via a network by a second communications agent

representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. In FIG. **4**, a content manager component **409** is illustrated as a component of communications agent **403**.

[0157] As described above, FIG. **7** illustrates first message **706**, including a data object identification request, received via network **506** by execution environment **401** of second node **504** from execution environment **401** of first node **502.**

[0158] In addition to packaging representations of data in a structure and/or format for sending a message according to a communications protocol, a content manager component **409** may operate to detect a data object identification request received and/or otherwise identified in a message received in a communication. Content manager component **409** operating in execution environment **401** of second node **504** may receive the message, illustrated by first message **706**, in a communication with execution environment **401** of first node **502**. Com-in component **411** in execution environment **401** of second node **504** may receive the message via communication protocol component **407** and network stack **405**. First message **706** may be delivered to execution environment **401** of second node **504** via network **506** based on a communications address of a communicant represented by communications agent **403** in execution environment **401** of second node **504**. A communications agent may represent more than one communicant, which may be identified by different addresses.

[0159] The first message **706** in FIG. **7** may be received in one or more packets via network **506** by network stack **405** and communications protocol component **407** in an instance of and/or analog of execution environment **401** including and/or otherwise provided by execution environment **401** of second node **504**. The data in the communication may be received by com-in component **411**. Com-in component **411** may provide the data to content manager component **409**. Content manager component **409** may determine one or more content types of the data. The content and/or portions of the content may be provided to one or more content type handler components **413** based on the one or more content types identified by content manager component **409**. For example, FIG. **8A** illustrates message portion **812a** including “text/plain” MIME type identifier **814a** as a content type identifier. Message portion **812a** may be provided to a

text/plain content handler **413**. Audio data in a voice communication may be provided to an audio content handler component **413**, and video data in a video communication may be provided to a video content handler component **413**.

[0160] Returning to FIG. **2B**, a block **214** illustrates that the method further includes detecting, by the second communications agent, a data object identification request included in the first message. Accordingly, a system for processing a data object identification request in a communication includes means for detecting, by the second communications agent, a data object identification request included in the first message. For example, the arrangement in FIG. **3B**, includes query handler component **302** that is operable for detecting, by the second communications agent, a data object identification request included in the first message. FIG. **4** illustrates query handler component **402** as an adaptation and/or analog of query handler component **302** in FIG. **3B**. One or more query handler components **402** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for detecting, by the second communications agent, a data object identification request included in the first message. In FIG. **4**, query handler component **402** is illustrated as a component of communications agent **403**.

[0161] FIG. **7** includes a dataflow message **708** that illustrates a detect data object identification request dataflow that may be internal to execution environment **401** of second node **504** and/or may include interoperation with another node via a network. With respect to FIG. **4**, dataflow message **708** may correspond to a data exchange and/or other interoperation between content handler component **409** and query handler component **402** to detect and/or otherwise process the data object identification request received in first message **706** by execution environment **401** of second node **504**. FIG. **7** further includes a dataflow message **710** that illustrates a search message for locating one or more data objects based on the data object identification request. The data object identification request may include and/or otherwise identify a data object matching criterion identified by data object information identified by the communicant represented by execution environment **401** of first node **502**. The one or more data

objects may be located and/or otherwise identified by determining a data object or data objects that match the data object matching criterion. The data object(s) may be in a data store accessible to execution environment **401** of second node **504**, such as one or more files in a file system of execution environment **401** of second node **504**. Dataflow message **710** may be internal to execution environment **401** of second node **504** and/or may be a message sent to a remote device, such as database server and/or a network attached storage device providing storage locations for some or all of a file system of execution environment **401** of second node **504**.

[0162] Com-in component **411** in execution environment **401** of second node **504**, as described above, provides the message, received from execution environment **401** of first node **502**, to content manager component **409**. Content manager component **409** is operatively coupled, via com-in component **411**, to communications protocol component **407** for receiving data in communications with other nodes, such as execution environment **401** of first node **502**. Thus, execution environment **401** of second node **504** may receive a message via a com-in component **411** in a communication with execution environment **401** of first node **502**. The message may include a data object identification request based on data object information identified by a user to communications agent **403** in execution environment **401** of first node **502**.

[0163] Data received in a communication, which may for example include a voice session and/or an instant message, may include data of various content types. In FIG. **4**, communications agent **403** in execution environment **401** of second node **504** includes one or more content handler components **413** to process data received according to its content type. Content manager component **409** may detect and/or otherwise identify a portion of the message as the data object identification request sent from execution environment **401** of first node **502**. The data object identification request may be identified by content manager component **409** according to a schema for the message and/or a schema for the data object identification request. For example, a portion of the message may include a MIME-type identifier that identifies the portion as a data object identification request and/or otherwise identifies the portion for routing to a query handler component **402** operating in and/or otherwise on behalf of execution environment **401** of second node **504**. Alternatively or additionally, the data object

identification request may have structure and/or content that is valid and identifiable according to a schema for defining and/or otherwise identifying valid data object identification requests. A query handler component, may be provided as a type of content handler component, in an aspect. Alternatively or additionally, a query handler component may interoperate with one or more content handler components to detect a data object identification request in a message received by a content manager component.

[0164] As such, query handler component **402** may operate along with other content handler components **413** to process data representations received in the message **706** from execution environment **401** of first node **502**. Query handler component **402** may also provide transformed data from the representations to one or more user interface element handler components **415**. Query handler component **402** operates to validate and process data object identification requests according to a schema defining valid data object identification requests. Various portions of a message including a data object identification request may be provided to one or more content handler components **413**, including query handler component **402**, to interoperate with presentation controller component **417** in execution environment **401** of second node **504** to present some or all of the received message, including the data object identification request.

[0165] FIG. **6D** illustrates a window, request window UI element **602d**, presenting an exemplary representation of a message received from execution environment **401** of first node **502** in a communication with second node **504**. As in FIG. **6A**, request window UI element **602d** includes a contactor UI element **604d**, a contactee UI element **606d**, a presentation space **608d** of the request window UI element **602d** that includes a user message UI element **610d** including a text portion of the message provided by the communicant represented by execution environment **401** of first node **502**. The various UI elements may be presented by one or more UI element handler components **415** in communications agent **403** operating in execution environment **401** of second node **504**. A reply UI element **648d** and a reply all UI element **648d** are illustrated as exemplary UI elements in presentation space **608d** presented to allow a user of execution environment **401** of second node **504** to respond and/or otherwise perform an

operation based on the received message. FIG. **6D** illustrates a data object identification request UI interface element **652d** where the data object identification request identified in the message from execution environment **401** of first node **502** is represented to the communicant represented by the execution environment **401** of second node **504**. In an aspect, the communicant is allowed to authorize the request by providing an input corresponding to a search and reply UI element **654d**. In another aspect, the communicant may be allowed to modify the received data object identification request.

[0166] FIG. **6E**, illustrates a reply window UI element **602e** that, in an aspect, may be presented by communications agent **403** in execution environment **401** of second node **504** in response to an user input corresponding to reply UI element **648d** detected by a corresponding UI element handler component **415**. In an aspect, a user of communications agent **403** operating in in execution environment **401** of second node **504** may provide input authorizing some or all of the processing of a data object identification request. Reply window UI element **602e** in FIG. **6E** includes a search UI element **656e** for receiving user input authorizing and/or otherwise instructing execution environment **401** of second node **504** to locate data objects that match the data object identification request, illustrated in a data object identification requested UI element **658e**.

[0167] As described herein in an aspect, content manager component **409** may detect content type information to detect a data object identification request in a communication. For example, the message portion illustrated in FIG. **8A** may be received in first message **706**. Content manager component **409** may detect "application/id-request" MIME type identifier in content-type header **804a**. The "application/id-request" MIME type identifier may be defined to identify a data object identification request based on one or more matching criteria for identifying a requested data object. Content manager component **409** may identify data object identification request portion **802a** as including the data object identification request.

[0168] In response to detecting a data object identification request in first message **706**, content manager component **409** may provide some or all of the data object identification request to query handler component **402**. For example, query handler

component **402** may be configured to operate according to a schema defining a format and/or a vocabulary for an XML-based language for id-request documents. Content manager component **409** may provide id-request document **806a**, as a data object identification request, to query handler component **402**. Query handler component **402** may operate according to the id-request schema. In an aspect, a query handler component may process more than one data object identification request content type. Alternatively or additionally, execution environment **401** may include multiple query handler components **402** for supporting multiple data object identification request content types.

[0169] Returning to FIG. **2B**, a block **216** illustrates that the method yet further includes generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. Accordingly, a system for processing a data object identification request in a communication includes means for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. For example, the arrangement in FIG. **3B**, includes query director component **312** that is operable for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. FIG. **4** illustrates query director component 412 as an adaptation and/or analog of query director component **312** in FIG. **3B**. One or more query director components **412** operate in an execution environment **401**. The system for query director includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is

identified by processing the data object identification request. In FIG. **4**, a query director component **412** is illustrated as a component of application **403**.

[0170] FIG. **7** includes dataflow message **712** that illustrates a create data object identification response data flow to process results returned from search dataflow **710**. In FIG. **4**, dataflow message **712** in FIG. **7** may represent data exchanged between query handler component **402** and query director component **412** to process results returned from a search operation performed based on the data object identification request, as illustrated by dataflow message **710**. Processing the results may include creating and/or otherwise constructing a data object identification response to the data object identification request identified in first message **706**. Dataflow message **712** may be internal to execution environment **401** of second node **504** and/or may include interoperation by execution environment **401** of second node **504** via a network with another node.

[0171] FIG. **6F**, illustrates an identify in response dialog UI element **660f** that, in an aspect, may be presented by communications agent **403** in execution environment **401** of second node **504** in response to processing the data object identification request to locate matching data objects. In an aspect, a user of communications agent **403** operating in execution environment **401** of second node **504** may provide input corresponding to search UI element **656e** in reply window UI element **602e** in FIG. **6E**. A corresponding UI element handler component **415** may operate to invoke query director component **412** to perform and/or otherwise to provide for performing one or more operations to locate one or more data objects based on the data object identification request. Query director component **412** may interoperate with one or more UI element handlers **415**, directly or indirectly via query handler component **402**, in an aspect, to present the UI element **660f** and its contents. Identify in response dialog UI element **660f** includes results pane UI element **662f** including representations of two data objects located and/or otherwise identified by query director component **412** based on the data object identification request. A user may select one or more data objects to identify in a data object identification response. An include UI element **664f** may be presented, as illustrated, to allow a user to instruct communications agent **403** to identify one or more selected data objects in results pane UI element **662f** in a data

object identification response. An include all UI element **666f** is illustrated to allow for user input instructing communications agent **403** to identify all data objects represented in the UI element **660f** in a data object identification response in a communication with execution environment **401** of first node **502**. A cancel UI element **668f** allows a user to end presentation of dialog UI element **660f**. A user may opt to identify no data objects represented in the dialog UI element **660f** and located and/or otherwise identified by query director component **412** based on the data object identification request.

[0172] Data objects located and/or otherwise identified by query director component **412** may be identified to IRB component **414** to construct and/or otherwise create a data object identification response identifying the data objects. As described above, in an aspect, a user may be allowed to select which data objects, if any, are to be identified in a data object identification response from one or more data objects located and/or otherwise identified by query director component **412** based on the data object identification response. In another aspect, communications agent **403** in FIG. **4** may invoke query director component **412** automatically to locate one or more data objects based on a data object matching criterion identified in a data object identification request and in response to detecting a data object identification request in a communication. In another aspect, communications agent **403** may invoke query director component **412** automatically to locate one or more data objects based on a data object identification request in a communication in response to detecting an indication to present, via a output device, some or all of a user message received in a message in the communication. Query director component **412** may be invoked and/or may operate based on the data object identification request and/or data object matching criterion identified to locate a data object prior to, during, and/or after other content received in the communication is processed.

[0173] The method illustrated in FIG. **2B** may include additional aspects supported by various adaptations and/or analogs of the arrangement of components in FIG. **4**. As described above, in one aspect a data object may be located and/or otherwise identified automatically in response to detecting the data object identification request. In another aspect, locating a data object may include presenting a locator UI element representing a data object identification request. Search UI element **656e** in FIG. **6E** exemplifies one

type of locator UI control. Search UI element **656e** may be presented by presentation controller component **417** as described above. A user input corresponding to a search UI element, such as search UI element **656e**, may be detected by presentation controller component **417**. Presentation controller component **417** may instruct query director component **412** to locate the data object based on a data object identification request represented by the data object identification request UI element **658e**.

[0174] In another aspect, a data object identification request UI element may be presented to receive one or more additional matching criteria from a user. Based on the data object identification request UI element and input from the user, query director component **412** may receive one or more additional data object matching criterion information and/or may receive input to modify a data object matching criterion identified in a received data object identification request. Query director component **412** may locate one or more data objects in response to receiving the additional criterion.

[0175] Returning to FIG. **2B**, a block **218** illustrates that the method yet further includes sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. Accordingly, a system for processing a data object identification request in a communication includes means for sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. For example, the arrangement in FIG. **3B**, includes identification response builder component **314** that is operable for sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. FIG. **4** illustrates identification response builder component **414** as an adaptation and/or analog of identification response builder component **314** in FIG. **3B**. One or more identification response builder components **414** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for sending a

second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. In FIG. **4**, a identification response builder component **414** is illustrated as a component of application **403**.

[0176] As described above, FIG. **7** includes second message dataflow message **714** illustrating second message sent via network **506** to execution environment **401** of first node **502** from execution environment **401** of second node **504**. Second message **714** includes a data object identification response in response to the data object identification request from first node **502** received and/or otherwise identified in first message **706**.

[0177] IRB component **414** may provide the data object identification response to content manager component **409** to include in a message, illustrated by second message **714**. As described above, content manager component **409** may receive and/or otherwise identify various portions of a message, including a data object identification response, from one or more content handler components **413**. The data object identification response may be provided to content manager component **409**. As described above, content manager component **409** in execution environment **401** of second node **504** may package the one or more data representations including a representation of the data object identification response into the message formatted according to a suitable communications protocol. Communications protocol component **407** operating in execution environment **401** of second node **504** may send the data as a message in a communication with execution environment **401** of first node **502** according to the specification(s) of the communications protocol. Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in the communication to the communications agent.

[0178] FIG. **8B** illustrates an exemplary data object identification response described in detail above. Second message portion **802b** illustrates a user message portion that may be included in second message **714**. Note that a user message portion may be empty or absent in some aspects. Data object identification response portion **804b** illustrates

the data object identification response as a list of URIs identifying data objects in a file system of execution environment **401** of second node **504**. User Message portion **802b** and data object identification response portion **804b** may be provided by respective content handler components **413** to content manager component **409**. Content manager component **409** may construct the content as illustrated in FIG. **8B** and/or otherwise provide for encoding, translating, combining, and/or otherwise preparing the content for sending in second message **714**.

[0179] The methods illustrated in FIG. **2A-B** may include additional aspects supported by various adaptations and/or analogs of the arrangement of components in FIG. **3A-B**. In various aspects, a data object identification request may be based on a date, a time, a length of time, a file type, a database record key, content of the data object, a content type identifier, a format rule, a vocabulary, a role of a user, a security attribute, a location in a data store such as a file system, an attribute of an identified data object, a size, a task, a transaction, a state, a user, a group, a requester, a relationship including a requesting user and a responding user, a keyword, a tag, a folder, and/or a path portion of a data object identifier – to name a few examples.

[0180] In another aspect, a data object located and/or otherwise identified in response to receiving a data object identification request must meet match a query identified by the data object identification request. Alternatively or additionally, a data object identification request may include an instruction and/or input for generating a data object. Locating the data object may include generating the data object. Generating a data object may include creating the data object and/or may include modifying and/or otherwise transforming an existing data object. For example, execution environment **401** of second node **504** may include a template stored in a file system. A data object identification request may identify the template. A query handler component **402** may operate to identify a document or other data object that may be created, is being created, and/or has been created based on the template. In response to an access request, execution environment **401** of second node **504** may return the document.

[0181] As described above, in various aspects, a data object identification request may be detected and/or represented based on various syntaxes, grammars, vocabularies, and/or languages. For example, a data object identification request may be identified

and/or represented according to a file system search syntax, a regular expression language, a structured query language (SQL) query, a universal data object identifier schema, an XPATH based language, an XQuery based language, an XML based language, an HTML based language (form-based), and/or a keyword-value pair based language.

[0182] A data object identification request in a communication may be communicated via a network according to a first communications protocol. A user message may be exchanged between communicants in the communication via a second communications protocol. For example, a data object identification request may be sent on behalf of a first communicant by execution environment **401** of first node **502** including a communications agent **403** representing the first communicant. The data object identification request may be included in a communication along with a text message sent to the execution environment **401** of second node **504**. The communicants represented by the execution environment **401** of first node **502** and the execution environment **401** of second node **504** may communicate via one or more exchanges of audio exchanged via, for example, a voice over IP (VoIP) communications protocol.

[0183] Exemplary data objects that may be requested via a data object identification request include a file, a program component, a data base record, video data, audio data, markup language, binary data, text data, an output of a service. Requested data objects may be pre-existing, volatile, and/or generated in response to the request.

[0184] As described above, receiving data object information may include receiving, via network from a node that received the data object information from the user, a message that identifies the data object information. For example, data object information may be received from a browser by a web server.

[0185] Data object information and/or a data object identification request may be generated, detected, and/or otherwise processed according to a schema that identifies at least one of a rule and a vocabulary that defines a valid data object identification request. Similarly, a data object identification response, an access request, and a response to an access request may be generated, detected, and/or otherwise processed according to a schema. One or more schemas may alone or together define one or more of valid data object information, a data object identification request, data

object identification response, an access request, and/or an access response. A schema, as just described, may be identified based on data received and/or otherwise identified by a node hosting a communications agent via the network from another node hosting a communications agent and/or from a path node included in a network path in the network communicatively coupling the node to one or more nodes that respectively host communications agents.

[0186] In a further aspect, multiple alternative schemas may be defined for each of data object information, data object identification requests, data object identification responses, access requests, an/or an access responses processing data object identification requests. A schema may be selected and/or otherwise identified from the multiple schemas by execution environment **401** of first node **502** and the execution environment **401** of second node **504** and/or based on information exchanged by the two nodes. For example, a schema may be selected by the first communicant and/or the second communicant. One may inform the other, via a communications protocol and/or via any other suitable means. The communicants may negotiate which schema to use via any suitable means.

[0187] The communications agents in execution environment **401** of second node **504** and execution environment **401** of first node **502** may select a schema For example, a schema may be identified and/or otherwise selected based on a communications protocol supported and/or a communications protocol not supported by one or both nodes. That is, in negotiating which communication protocol to use allowing communicants to exchange information, a schema may be identified. A schema may be identified and/or otherwise selected based on an attribute of the data object. A schema for identifying image data may be different than a schema for editable documents. A schema for identifying a data object in a first file system may be different than a schema for identifying a data object in a different file system.

[0188] As described above, data object information and/or a data object identification request may identify a scope that specifies all or some portion of one or more data stores for locating a data object. For example, A communications agent **403** in execution environment **401** of first node **502** may receive user input that identifies a

folder in a file system of the execution environment **401** of second node **504** that defines a scope for locating a data object.

[0189] Data object information and/or a data object identification request may be represented in a representation that includes a complete or a portion of at least one of a keyword expression, a regular expression, an XQuery expression, an XPath expression, a file system path expression, and a structured query language statement.

[0190] Further as also described above, a data object identification request, a data object identification response, an access request, an/or an access response may be included in a message according to a schema that identifies the message as valid for a specified communications protocol. Those skilled in the art are aware that schemas for email differ from schemas for instant messages and from streamed audio, for example.

[0191] A data object identification request, a data object identification response, an access request, and/or an access response may be exchanged in a communication in which audio data and/or image data is also exchanged. Multimedia may be a packaged to include one or more data streams which may include text based data streams.

[0192] A data object identification request, a data object identification response, an access request, and/or an access response may be included in an email, an instant message, a multi-media message, a short-message service message, and/or a data stream. The data stream may include audio data, image data, and/or text data.

[0193] A data object identification request, a data object identification response, an access request, an/or an access response may be included in and/or detected in a communication based on a location in the communication and/or based on a marker, such as an identifier in the communication that identifies a location for the data object identification request, the data object identification response, the access request, and/or the access response. The location may be absolute, such as fixed number of bytes or fields from the start or end of a message. Alternatively or additionally, the location may be identified by a location relative to another detectable portion of the communication. FIGs. **8A-B** illustrate the use of MIME type identifiers as markers that identify locations for one or more of a data object identification request, a data object identification response, an access request, an/or an access response in a communication.

[0194] A data object identification request, a data object identification response, an access request, an/or an access response or respective portions thereof may include at least a portion of a keyword expression, a regular expression, expression including a Boolean operator, an expression including a precedence information, and a structured query language statement.

[0195] With respect to the method illustrated in FIG. **2B**, the method as described above may include locating and/or otherwise identifying a data object automatically in response to receiving an access request that matches and/or otherwise identifies the data object. Further, a data object identification response may be created and/or otherwise generated automatically, in response to locating and/or otherwise identifying the data object.

[0196] In an aspect, in response to receiving and/or otherwise detecting a data object identification request, a user interface may be presented based on the data object identification request to prompt a user to process the data object identification request in some manner, such as authorizing the request and/or modifying the request by restricting its scope. In response to detecting user input targeting and/or otherwise corresponding to the presented user interface, one or more data objects may be located and/or otherwise identified based on the data object identification request.

[0197] A user may change a received data object identification request. One or more data objects may be located and/or otherwise identified based on the changed data object identification request.

[0198] In a further aspect, once one or more data objects are located and/or otherwise identified based on a data object identification request, a user interface may be presented to allow a user to identify the one or more data objects. User information may be received that selects one or more data objects to identify in a data object identification response and/or to not identify in the data object identification response. A user, in an aspect, may be allowed to identify a data object not located and/or otherwise identified by the user's node. The data object, based on the user input, is identified in the data object identification response.

[0199] In light of the description provided above, a data object identification response may be generated automatically in response to detecting a corresponding data object

identification request by an execution environment **401** of first node **502**. Further, the automatically generated data object identification response, may be sent to the node that sent the corresponding data object identification request, automatically in response to the generating of the data object identification response.

[0200] With respect to the method illustrated in FIG. **2A**, the third message may be sent automatically in response to detecting the data object identification response. In another aspect, a representation of a received data object identification response may be presented, via an output device, to a user. The representation may identify one or more data objects located and/or otherwise identified based on the data object identification request. The representation may be presented automatically, in response to detecting the data object identification response. Further, a user input may be detected that corresponds to and/or otherwise targets the representation. The user input may be processed as an indication to retrieve the data object. As a result an access request may be created or otherwise generated that identifies the data object. The access request may be sent in a message via the network to retrieve the data object. The message may be sent to a node that sent the data object identification response and/or to another node. The third message may be sent by the communications agent in the execution environment **401** of first node **502** to the communications agent in the execution environment **401** of second node **504** via a communications protocol. The third message sent via the communication protocol may include a communications address, in an address space of the communications protocol, that identifies a user represented by the communications agent in the execution environment **401** of second node **504**. The third message may include a data object identification request, a data object identification response, and/or a data object sent in response to previously receiving an access response by the execution environment **401** of first node **502**. The third message may include an access request that identifies the data object. The access request may include a universal data object identifier (URI) that identifies the data object.

[0201] With respect to the method illustrated in FIG. **2B**, in response to receiving an access request, a data object identified in the request may be sent. The data object may be sent automatically, in response to receiving the access request. The data object may

be sent as attachment in a communication. The data object may be sent by a communications agent in the execution environment **401** of second node **504** to a communications agent in the execution environment **401** of first node **502** via a communications protocol. The data object may be sent via a communication that includes a communications address, in an address space of the communications protocol, that identifies the first communicant represented by the execution environment **401** of first node **502**. The communication that includes the data object may include a data object identification request, a data object identification response, and/or an access request sent in response to a data object identification response sent previously by the first node.

[0202] The methods illustrated in FIGs. **2A-B** may yet further include additional aspects supported by various adaptations and/or analogs of the arrangement of components in FIGs. **3A-B**, respectively.

[0203] Performing the method illustrated in FIG. **2A** and/or the method illustrated in FIG. **2B** and/or any of its extension and/or in any of its aspects may include one or more of calling a function or method of an object, sending a message via a network; sending a message via an inter-process communication mechanism such as a pipe, a semaphore, a shared data area, and/or a queue; and/or receiving a request such as poll and responding to invoke, and sending an asynchronous message.

[0204] Output devices suitable for presenting a representation of a data object and/or a tag include a visual output device, an audio output device, and a tactile output device. One output device may present a data object and another output device may present a tag with which the data object is tagged.

[0205] Any action included in performing the method illustrated in FIG. **2A** as well as any action included in in performing the method illustrated in FIG. **2B** may include sending and/or receiving a message via a network. Further, a message, included in performing any of the subject matter described herein and/or any of its extensions in any of its aspects, may an asynchronous message without a corresponding request.

[0206] To the accomplishment of the foregoing and related ends, the descriptions and annexed drawings set forth certain illustrative aspects and implementations of the disclosure. These are indicative of but a few of the various ways in which one or more

aspects of the disclosure may be employed. The other aspects, advantages, and novel features of the disclosure will become apparent from the detailed description included herein when considered in conjunction with the annexed drawings.

[0207] It should be understood that the various components illustrated in the various block diagrams represent logical components that are configured to perform the functionality described herein and may be implemented in software, hardware, or a combination of the two. Moreover, some or all of these logical components may be combined, some may be omitted altogether, and additional components may be added while still achieving the functionality described herein. Thus, the subject matter described herein may be embodied in many different variations, and all such variations are contemplated to be within the scope of what is claimed.

[0208] To facilitate an understanding of the subject matter described above, many aspects are described in terms of sequences of actions that may be performed by elements of a computer system. For example, it will be recognized that the various actions may be performed by specialized circuits or circuitry (e.g., discrete logic gates interconnected to perform a specialized function), by program instructions being executed by one or more processors, or by a combination of both. The description herein of any sequence of actions is not intended to imply that the specific order described for performing that sequence must be followed.

[0209] Moreover, the methods described herein may be embodied in executable instructions stored in a non-transitory computer readable medium for use by or in connection with an instruction execution machine, system, apparatus, or device, such as a computer-based or processor-containing machine, system, apparatus, or device. As used here, a "non-transitory computer readable medium" may include one or more of any suitable media for storing the executable instructions of a computer program in one or more forms including an electronic, magnetic, optical, and electromagnetic form, such that the instruction execution machine, system, apparatus, or device may read (or fetch) the instructions from the non-transitory computer readable medium and execute the instructions for carrying out the described methods. A non-exhaustive list of conventional exemplary non-transitory computer readable media includes a portable computer diskette; a random access memory (RAM); a read only memory (ROM); an

erasable programmable read only memory (EPROM or Flash memory); optical storage devices, including a portable compact disc (CD), a portable digital video disc (DVD), a high definition DVD (HD-DVD.TM.), and a Blu-ray.TM. disc; and the like

[0210] Thus, the subject matter described herein may be embodied in many different forms, and all such forms are contemplated to be within the scope of what is claimed. It will be understood that various details may be changed without departing from the scope of the claimed subject matter. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation, as the scope of protection sought is defined by the claims as set forth hereinafter together with any equivalents.

[0211] Thus, the subject matter described herein may be embodied in many different forms, and all such forms are contemplated to be within the scope of what is claimed. It will be understood that various details may be changed without departing from the scope of the claimed subject matter. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation, as the scope of protection sought is defined by the claims as set forth hereinafter together with any equivalents.

[0212] All methods described herein may be performed in any order unless otherwise indicated herein explicitly or by context. The use of the terms "a" and "an" and "the" and similar referents in the context of the foregoing description and in the context of the following claims are to be construed to include the singular and the plural, unless otherwise indicated herein explicitly or clearly contradicted by context. The foregoing description is not to be interpreted as indicating that any non-claimed element is essential to the practice of the subject matter as claimed.

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION
First Named Inventor/Applicant Name:	Robert Paul Morris
Filer:	Patrick Edgar Caldwell
Attorney Docket Number:	DR0149C-CIP

Filed as Small Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
UTILITY FILING FEE (ELECTRONIC FILING)	4011	1	75	75
UTILITY SEARCH FEE	2111	1	330	330
UTILITY EXAMINATION FEE	2311	1	380	380
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
			Total in USD ($)	**785**

Electronic Acknowledgement Receipt

EFS ID:	32671029
Application Number:	15984406
International Application Number:	
Confirmation Number:	3031
Title of Invention:	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION
First Named Inventor/Applicant Name:	Robert Paul Morris
Customer Number:	92045
Filer:	Patrick Edgar Caldwell
Filer Authorized By:	
Attorney Docket Number:	DR0149C-CIP
Receipt Date:	20-MAY-2018
Filing Date:	
Time Stamp:	22:12:01
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$785
RAM confirmation Number	052118INTEFSW22124000
Deposit Account	
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	DR0149C-CIP_ADS_vF_20-May-2018.pdf	1822942 f0ce05f589a607c6029213f6731a791584bec8ba	no	8
Warnings:					
Information:					
2	Oath or Declaration filed	DR0149C-CIP_Declaration_SIGNED_vF.PDF	150675 8668cc715124c840bbc8b25a37abe364b014e6d0	no	2
Warnings:					
Information:					
3	Drawings-only black and white line drawings	DR0149C-CIP_Drawings_vF_20-May-2018.pdf	1712001 96f6658469a446d49ab31b0f689d47cf23273e9e	no	15
Warnings:					
Information:					
4	Power of Attorney	DR0149C-CIP_POA_SittingMan_vF.pdf	159031 1906a0fadfe1bac9befaa266bbf1bdad44203787	no	1
Warnings:					
Information:					
5		DR0149C-CIP_App_vF_20-May-2018.pdf	422481 6d3a192ffe57baedde73d1ffffad5da2d782455b	yes	91

Multipart Description/PDF files in .zip description

Document Description	Start	End
Abstract	91	91
Claims	75	90
Specification	1	74

Warnings:					
Information:					
6	Fee Worksheet (SB06)	fee-info.pdf	35507 70ef6ab43d45e882ebe6c8bd18bfb6048b7498d9	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	4302637	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149C-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2:

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1 — Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Robert	Paul	Morris	

Residence Information (Select One) ● US Residency Non US Residency Active US Military Service

City	State/Province	Country of Residence
Raleigh	NC	US

Mailing Address of Inventor:

Address 1	712 Latta Street
Address 2	
City	Raleigh
State/Province	NC
Postal Code	27607
Country	US

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	92045		
Email Address	pcaldwell@thecaldwellfirm.com	Add Email	Remove Email

Application Information:

Title of the Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION		
Attorney Docket Number	DR0149C-CIP	**Small Entity Status Claimed**	☒
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)		**Suggested Figure for Publication (if any)**	

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149C-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☒ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	● Customer Number	US Patent Practitioner	◯ Limited Recognition (37 CFR 11.9)
Customer Number	92045		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.
When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
	Continuation of	15724223	2017-10-03

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149C-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15724223	Continuation in part of	14274623	2014-05-09
Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
14274623	Continuation in part of	13624906	2012-09-22
Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
14274623	Continuation in part of	13626635	2012-09-25

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button. Add

Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)[i] the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Remove

Application Number	Country[i]	Filing Date (YYYY-MM-DD)	Access Code[i] (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button. Add

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149C-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☒ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149C-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Applicant 1 — Remove

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Clear

● Assignee	Legal Representative under 35 U.S.C. 117	Joint Inventor
Person to whom the inventor is obligated to assign.	Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor:

If the Applicant is an Organization check here. ☒

Organization Name	Sitting Man, LLC		
Mailing Address Information For Applicant:			
Address 1	712 Latta Street		
Address 2			
City	Raleigh	**State/Province**	NC
Country	US	Postal Code	27607
Phone Number		Fax Number	
Email Address			

Additional Applicant Data may be generated within this form by selecting the Add button. Add

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149C-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

Remove

If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1			
Address 2			
City		**State/Province**	
Country i		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button. Add

Signature:

Remove

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). **However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).**

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/Patrick Caldwell/			Date (YYYY-MM-DD)	2018-05-20
First Name	Patrick	Last Name	Caldwell	Registration Number	44580

Additional Signature may be generated within this form by selecting the Add button. Add

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149C-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1 The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3 A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent CooperationTreaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

PTO/AIA/01 (06-12)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION

As the below named inventor, I hereby declare that:

This declaration is directed to:

- [■] The attached application, or
- [ ] United States application or PCT international application number _____

filed on _____.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Robert Paul Morris Date (Optional) :_____

Signature: /Robert Paul Morris/

Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form or must have been previously filed. Use an additional PTO/AIA/01 form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 1 minute to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

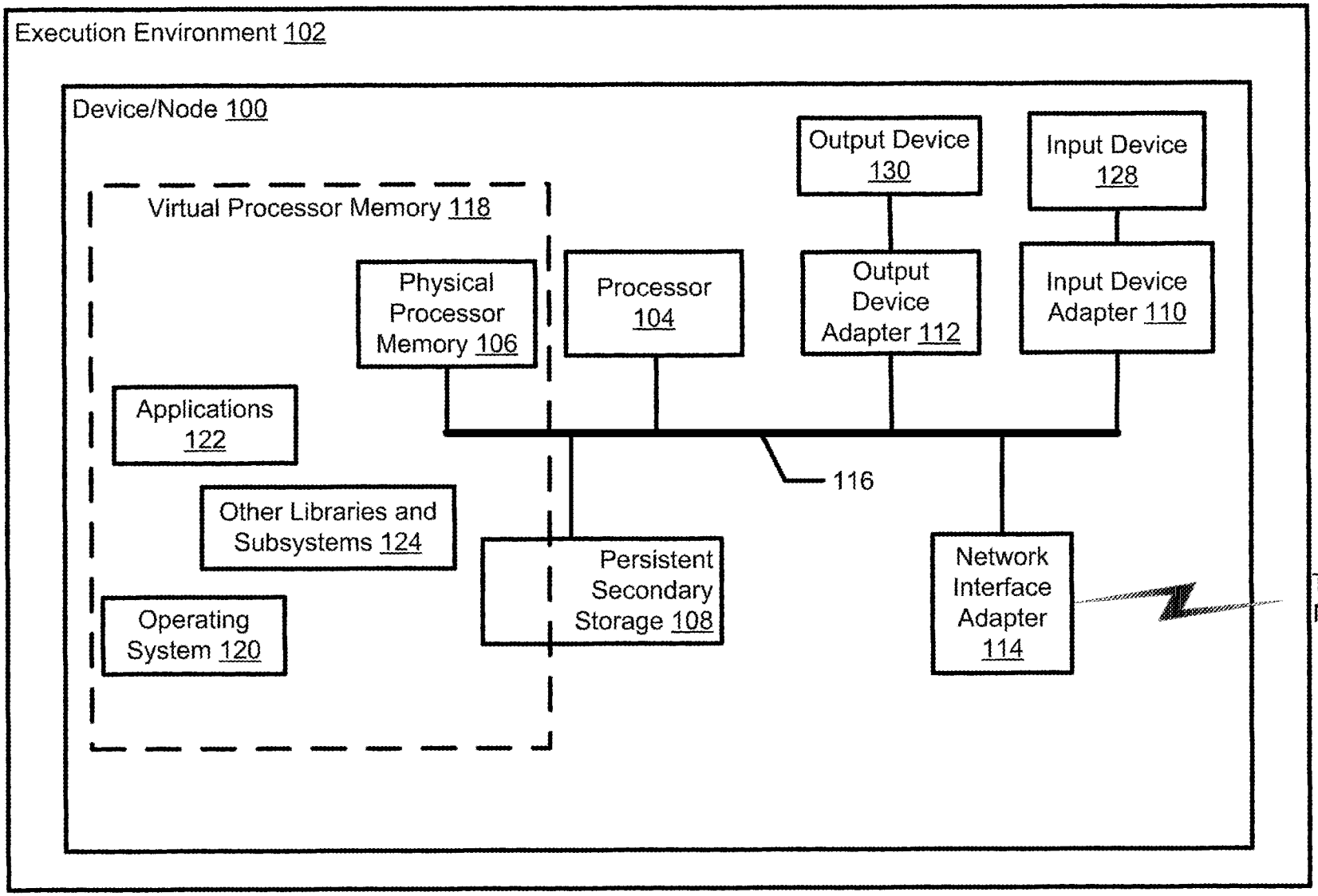

Execution Environment 102
Device/Node 100
Virtual Processor Memory 118
Physical Processor Memory 106
Processor 104
Output Device 130
Input Device 128
Output Device Adapter 112
Input Device Adapter 110
Applications 122
Other Libraries and Subsystems 124
Operating System 120
Persistent Secondary Storage 108
116
Network Interface Adapter 114
To/From Network

FIG. 1

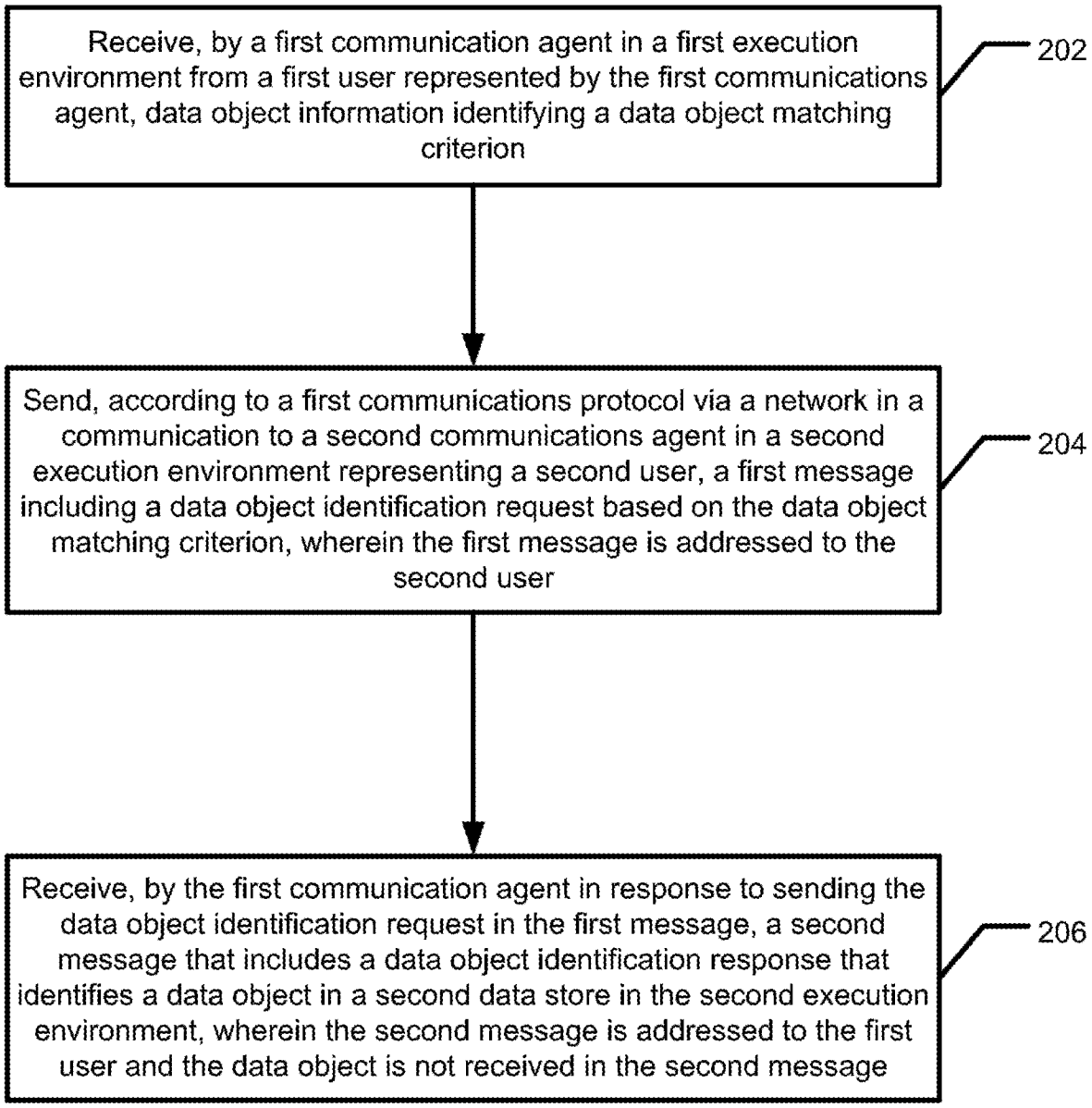

Receive, by a first communication agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion
202
Send, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user
204
Receive, by the first communication agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message
206

FIG. 2A

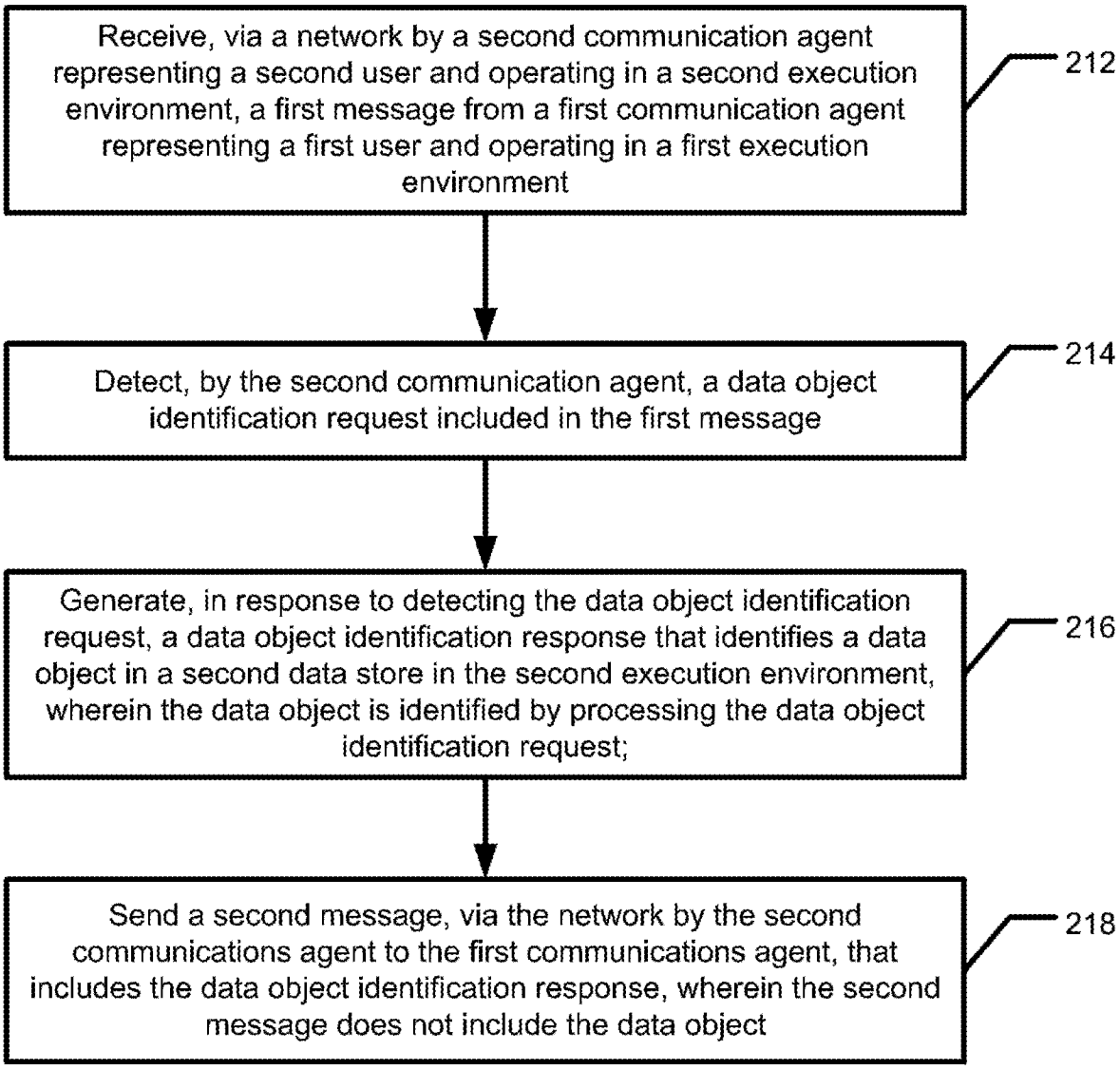

Receive, via a network by a second communication agent representing a second user and operating in a second execution environment, a first message from a first communication agent representing a first user and operating in a first execution environment
212
Detect, by the second communication agent, a data object identification request included in the first message
214
Generate, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request;
216
Send a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object
218

FIG. 2B

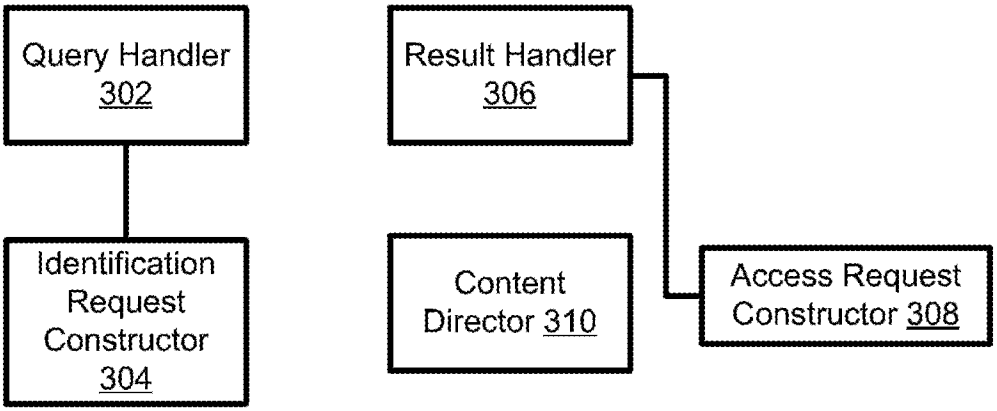

Query Handler 302
Result Handler 306
Identification Request Constructor 304
Content Director 310
Access Request Constructor 308

FIG. 3A

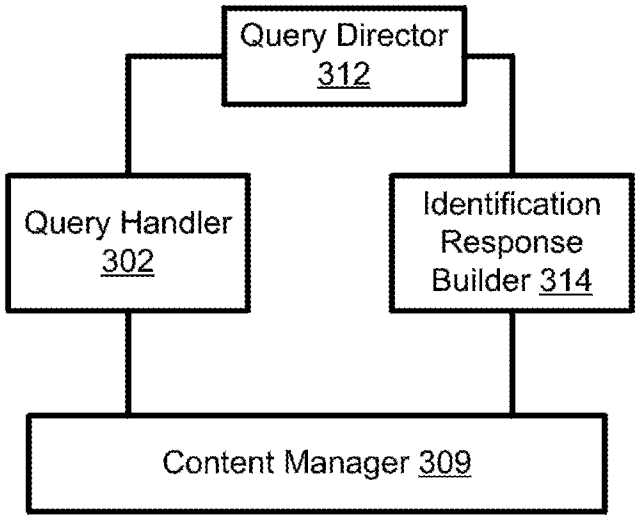

Query Director 312
Query Handler 302
Identification Response Builder 314
Content Manager 309

FIG. 3B

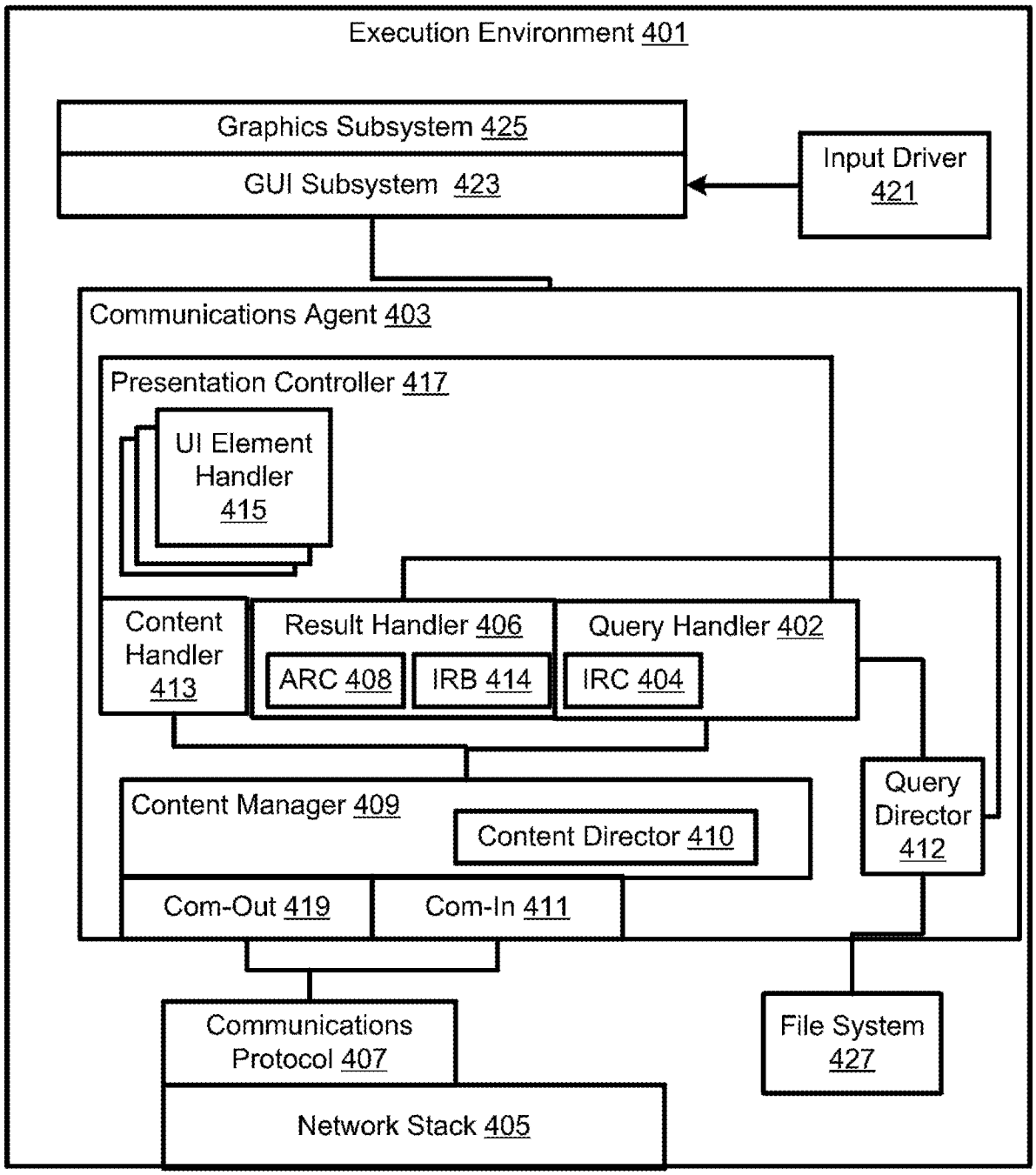

Execution Environment 401
Graphics Subsystem 425
GUI Subsystem 423
Input Driver 421
Communications Agent 403
Presentation Controller 417
UI Element Handler 415
Content Handler 413
Result Handler 406
ARC 408
IRB 414
Query Handler 402
IRC 404
Content Manager 409
Content Director 410
Query Director 412
Com-Out 419
Com-In 411
Communications Protocol 407
Network Stack 405
File System 427

FIG. 4

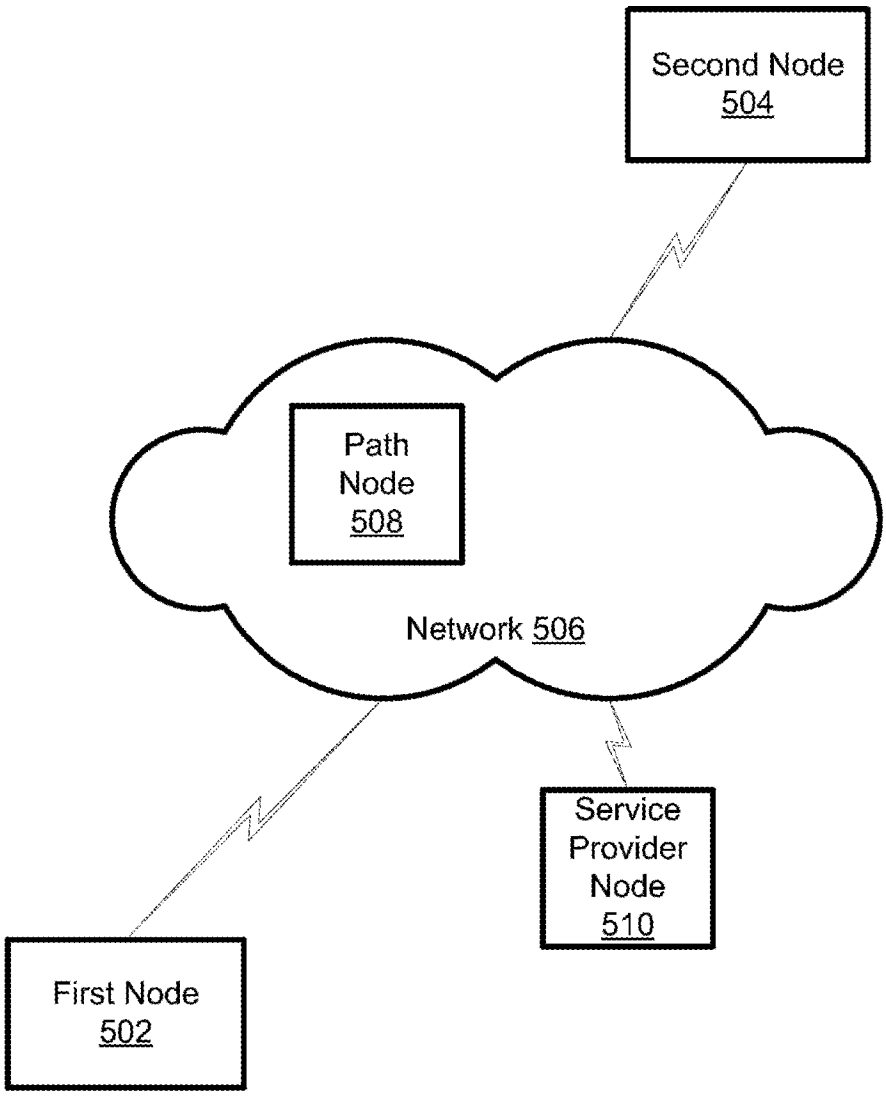

Second Node
504
Path
Node
508
Network 506
Service
Provider
Node
510
First Node
502

FIG. 5

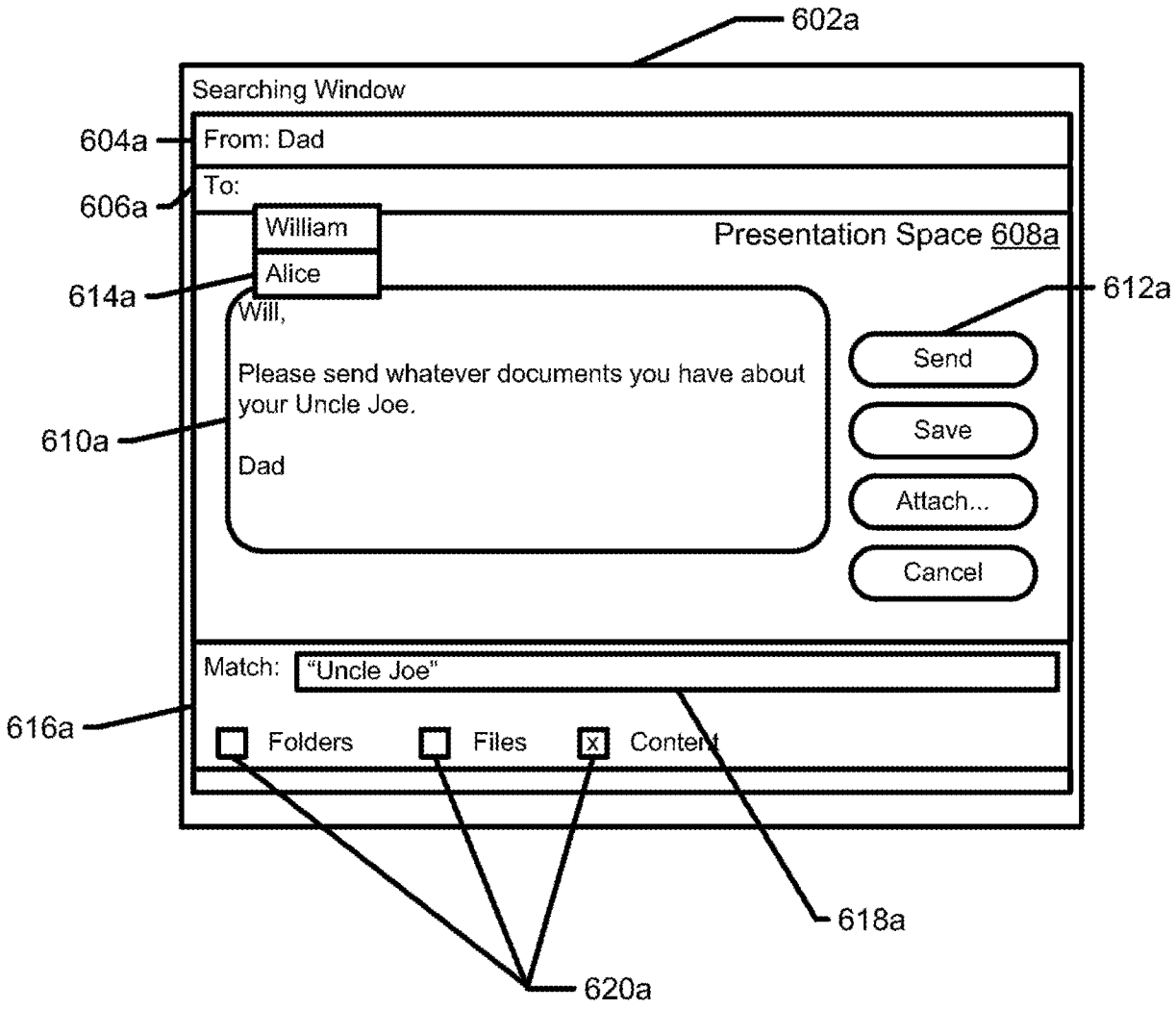

602a
Searching Window
604a
From: Dad
606a
To:
William
Alice
614a
Presentation Space 608a
612a
Will,
Please send whatever documents you have about your Uncle Joe.
610a
Dad
Send
Save
Attach...
Cancel
Match:
"Uncle Joe"
616a
Folders
Files
Content
618a
620a

FIG. 6A

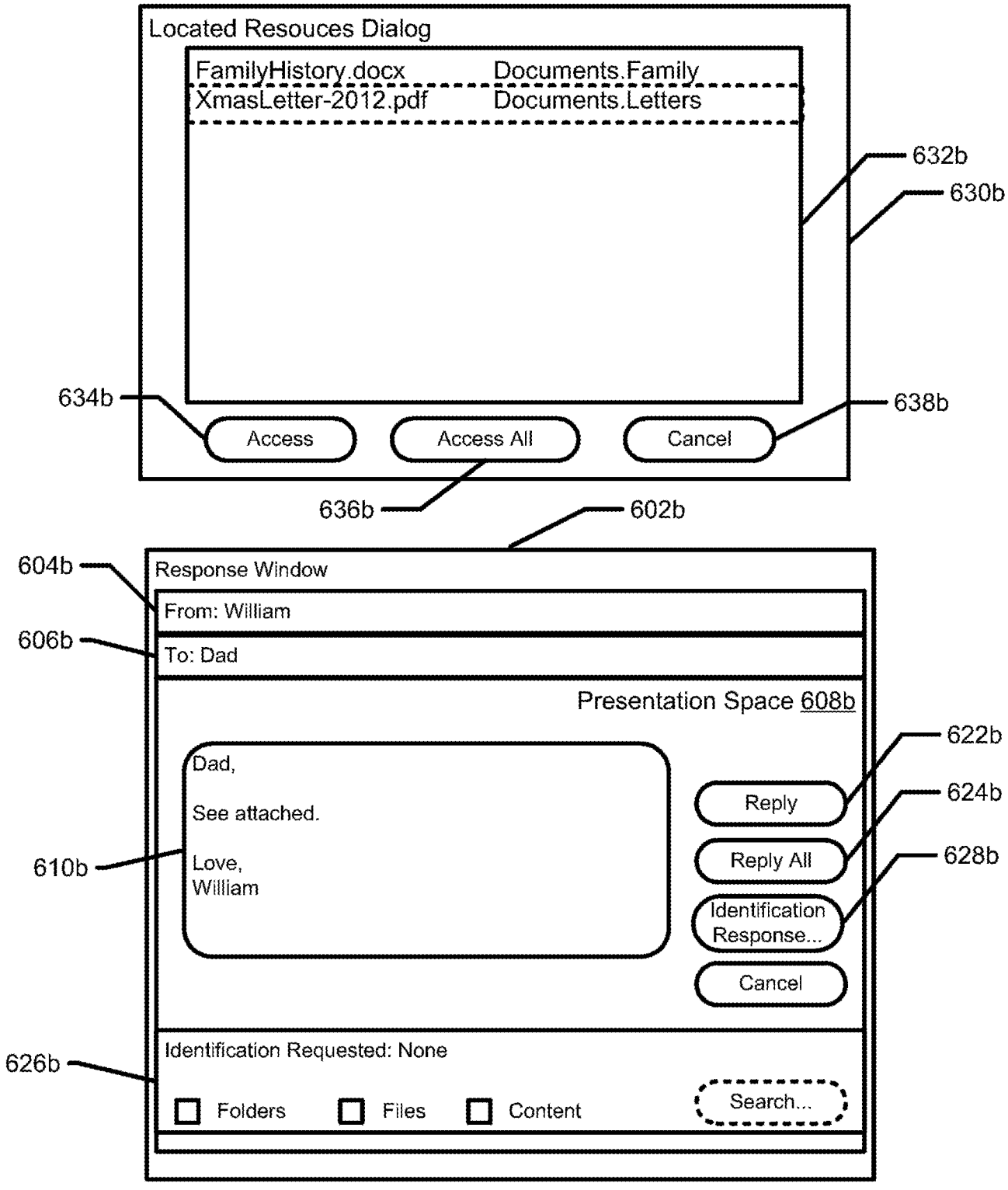

Located Resouces Dialog
FamilyHistory.docx Documents.Family
XmasLetter-2012.pdf Documents.Letters
632b
630b
634b
638b
Access
Access All
Cancel
636b
602b
604b
Response Window
From: William
606b
To: Dad
Presentation Space 608b
622b
624b
628b
610b
Dad,
See attached.
Love,
William
Reply
Reply All
Identification Response...
Cancel
626b
Identification Requested: None
Folders
Files
Content
Search...

FIG. 6B

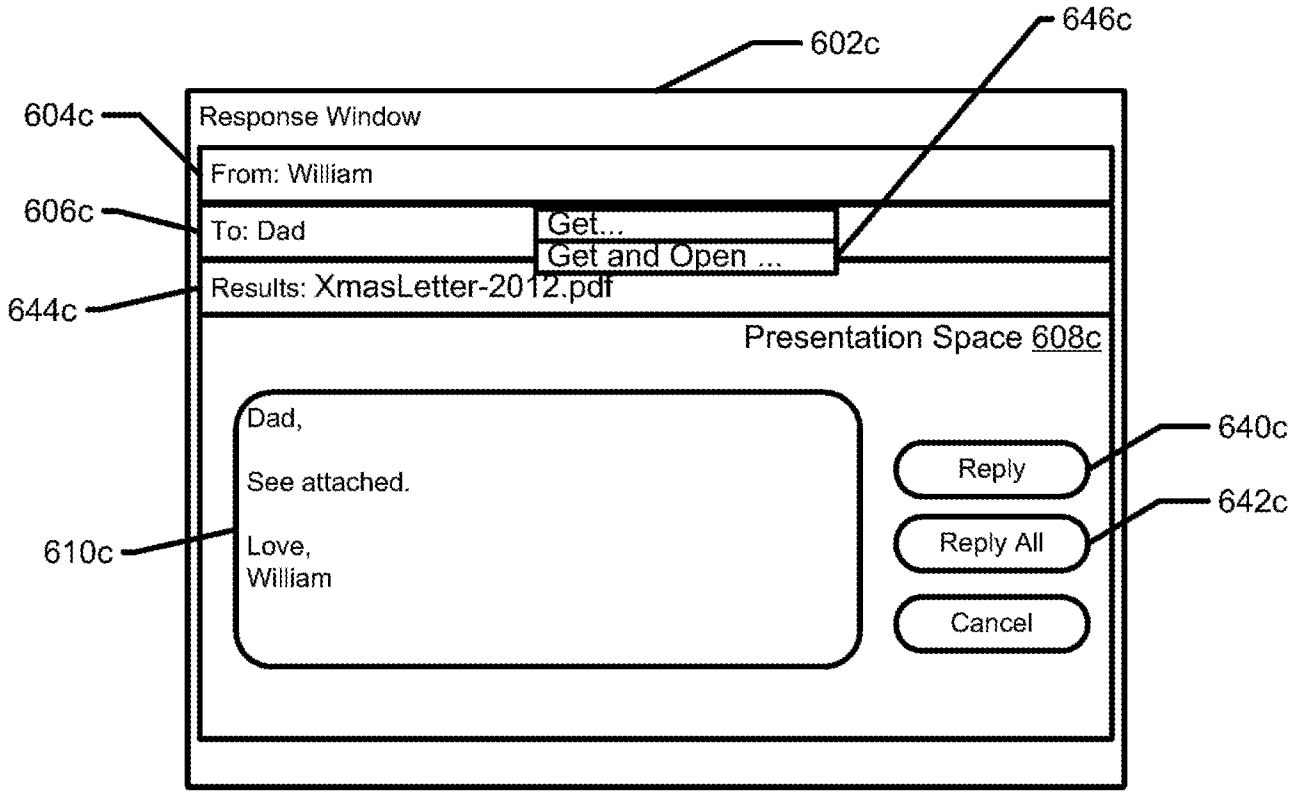

602c
646c
604c
Response Window
From: William
606c
To: Dad
Get...
Get and Open ...
644c
Results: XmasLetter-2012.pdf
Presentation Space 608c
Dad,
See attached.
Love,
William
610c
Reply
640c
Reply All
642c
Cancel

FIG. 6C

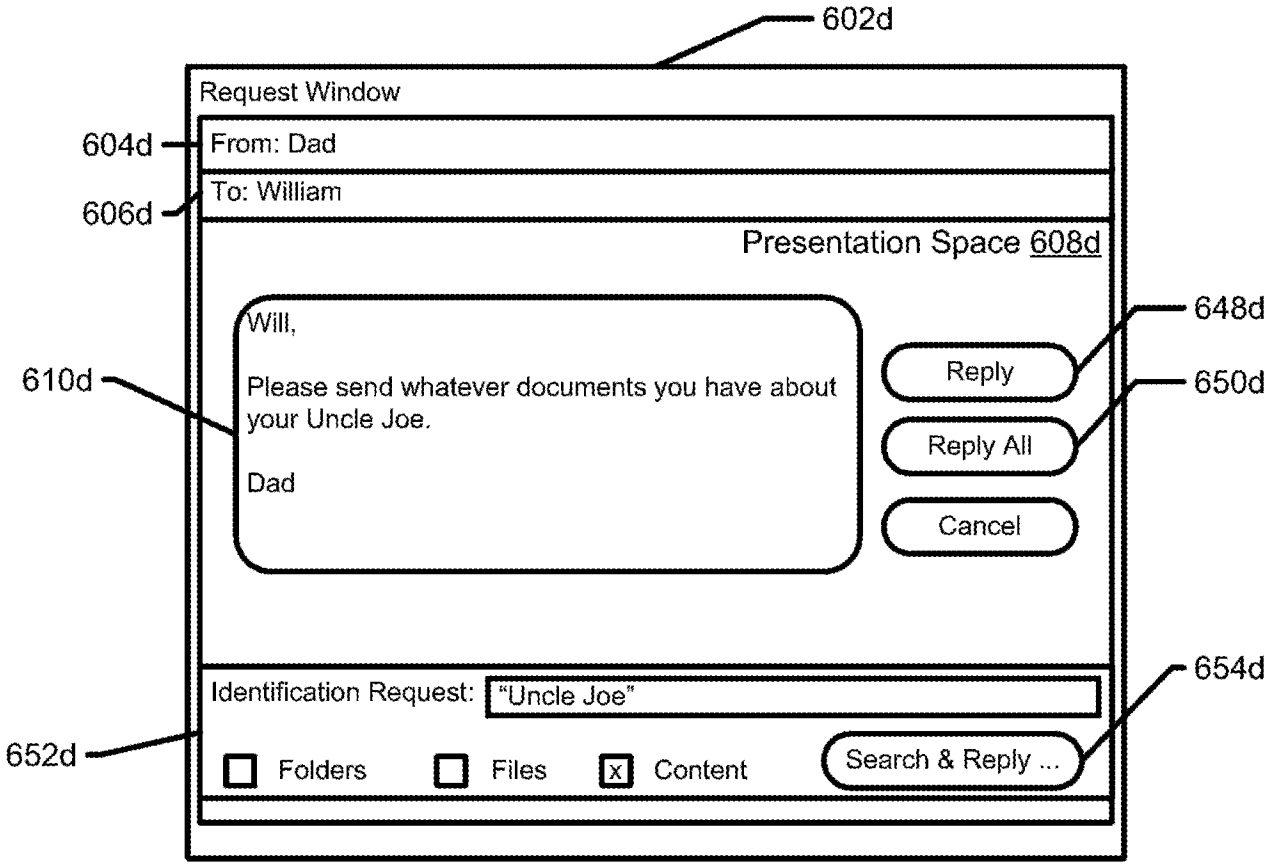

602d
Request Window
604d
From: Dad
606d
To: William
Presentation Space 608d
610d
Will,
Please send whatever documents you have about your Uncle Joe.
Dad
648d
Reply
650d
Reply All
Cancel
654d
Identification Request:
"Uncle Joe"
652d
Folders
Files
x
Content
Search & Reply ...

FIG. 6D

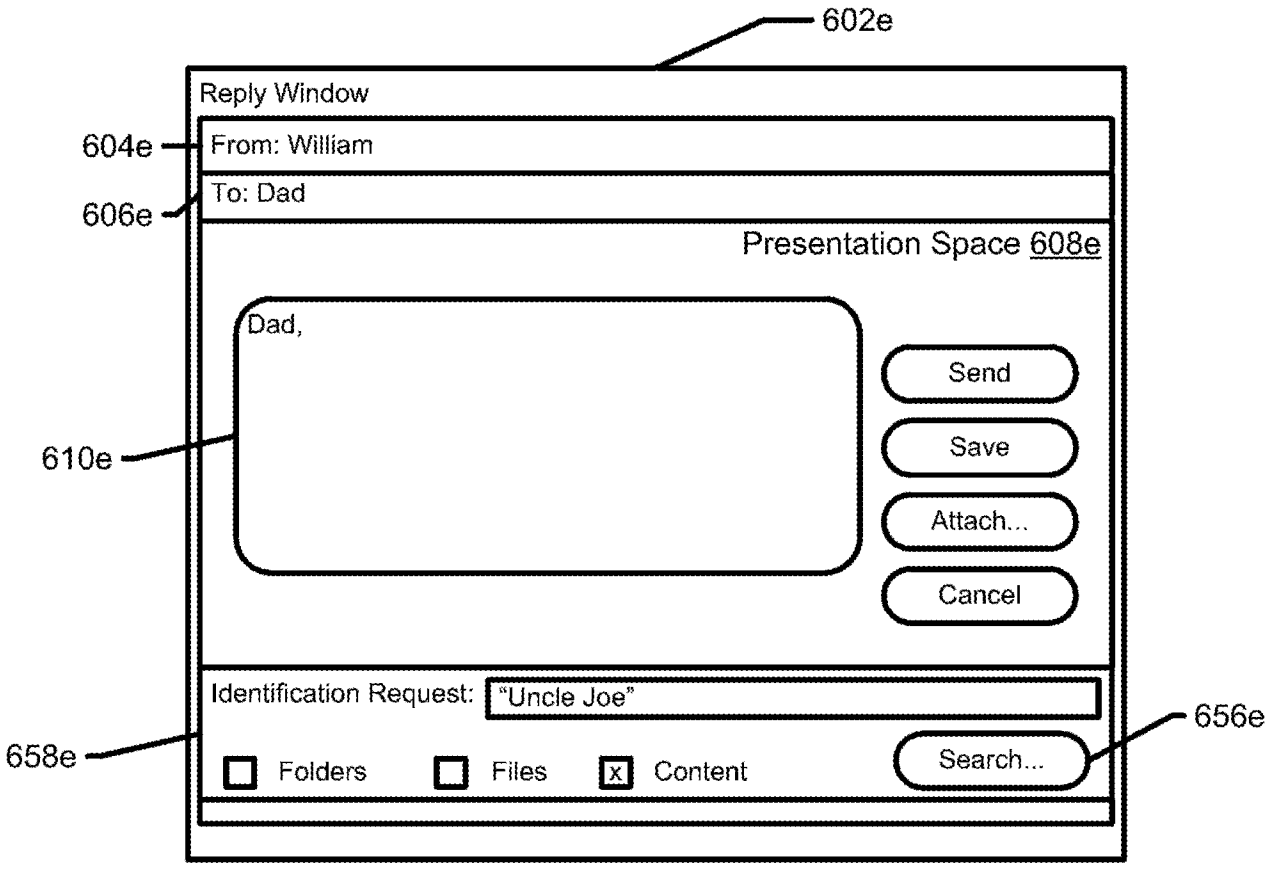

602e
Reply Window
604e
From: William
606e
To: Dad
Presentation Space 608e
Dad,
610e
Send
Save
Attach...
Cancel
Identification Request:
"Uncle Joe"
658e
656e
Folders
Files
x
Content
Search...

FIG. 6E

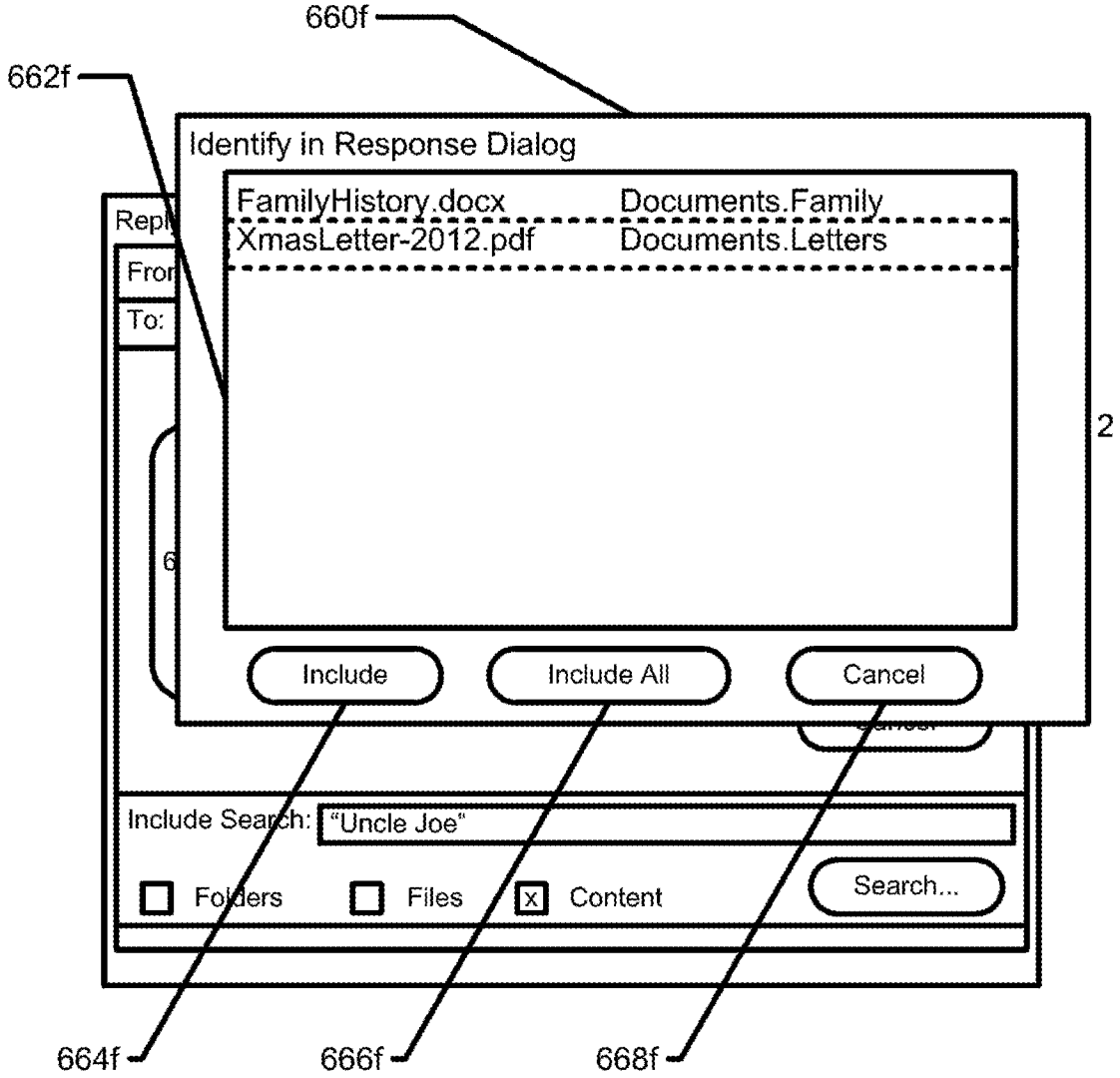

660f
662f
Identify in Response Dialog
FamilyHistory.docx Documents.Family
XmasLetter-2012.pdf Documents.Letters
Include
Include All
Cancel
Include Search: "Uncle Joe"
Folders
Files
Content
Search...
664f
666f
668f

FIG.6F

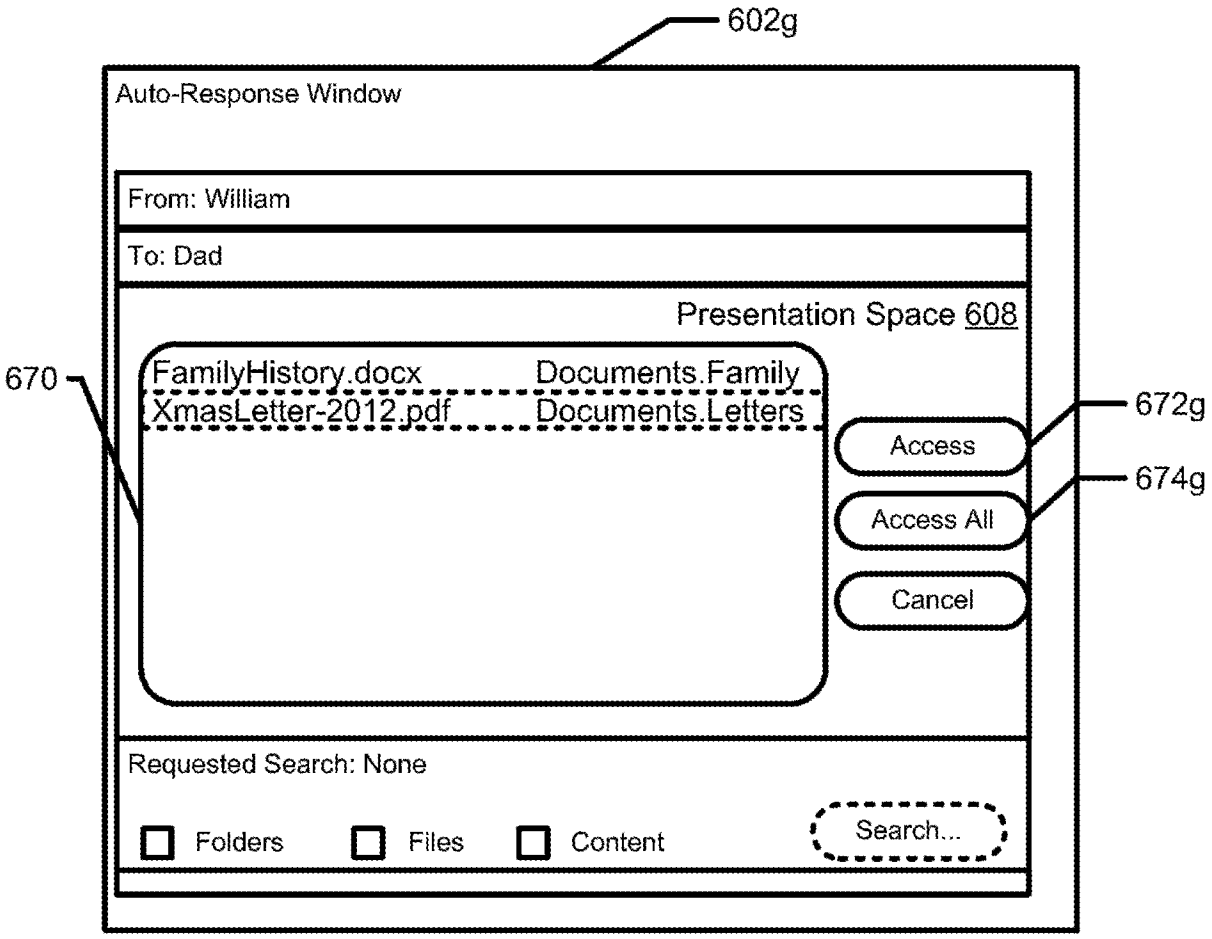

602g
Auto-Response Window
From: William
To: Dad
Presentation Space 608
670
FamilyHistory.docx Documents.Family
XmasLetter-2012.pdf Documents.Letters
Access
Access All
Cancel
672g
674g
Requested Search: None
Folders
Files
Content
Search...

FIG. 6G

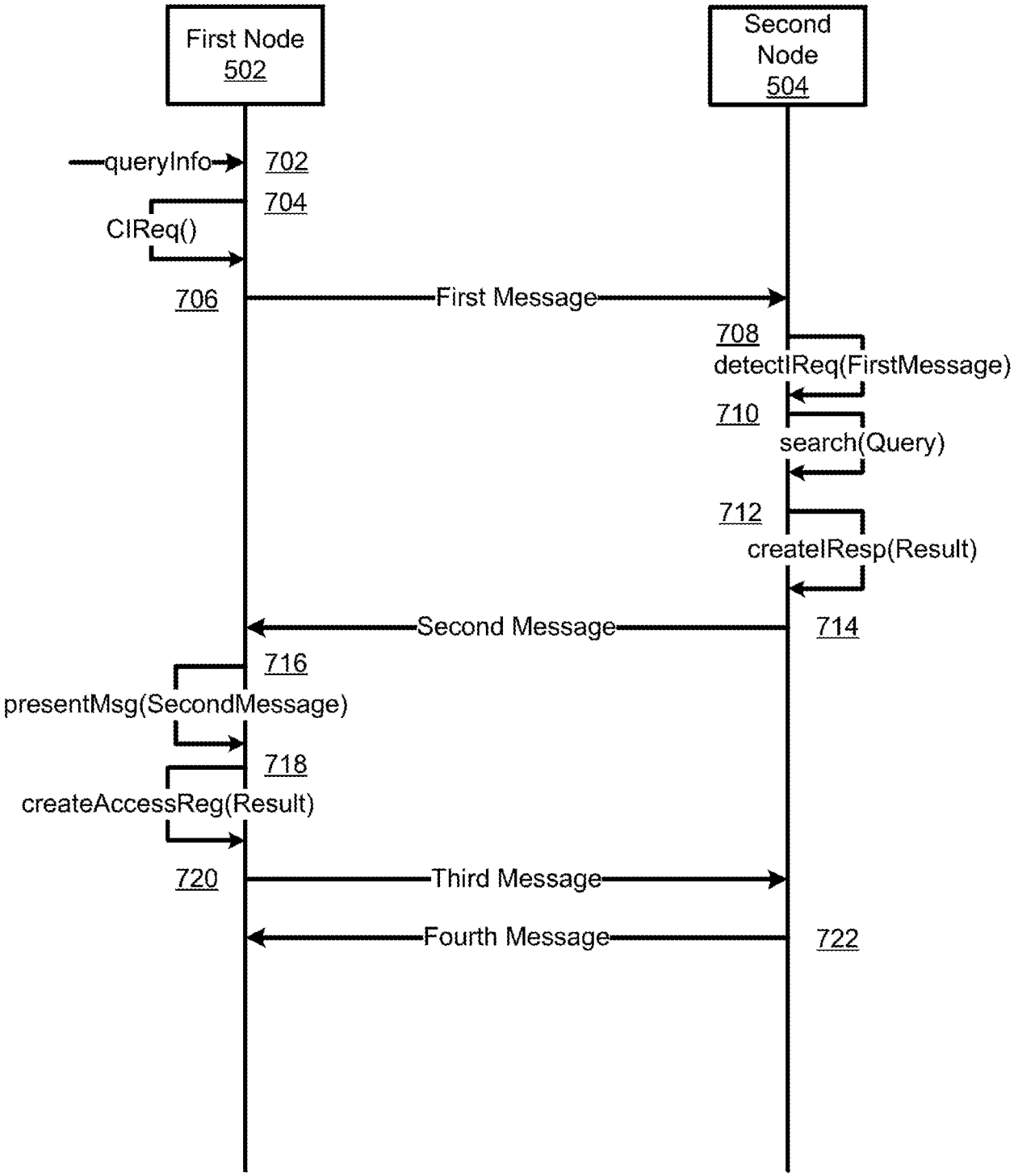

First Node
502
Second
Node
504
queryInfo
702
704
CIReq()
706
First Message
708
detectIReq(FirstMessage)
710
search(Query)
712
createIResp(Result)
Second Message
714
716
presentMsg(SecondMessage)
718
createAccessReg(Result)
720
Third Message
Fourth Message
722

FIG. 7

800a

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="--OoOT----toOo--------"

--OoOT----toOo--------
Content-Type: text/plain
Hi, this is the first message
--OoOT----toOo--------
Content-Type: application/id-request
Accept:  image/*
<id-request>
  <and>
    <criterion attribute="query" type="keyword" match-expression="'Uncle Joe'" />
    <criterion attribute="scope" resource-type="content" />
  </and>
</id-request>
--OoOT----toOo--------
```

814a, 812a, 804a, 806a, 810a, 808a, 802a

Fig. 8A

800b

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="--toOo----OoOt--------"

--toOo----OoOt---------
Content-Type: text/plain
Dad,

See attached.

Love,
William
--toOo----OoOt---------
Content-Type: application/query-respponse

FILE://Documents/Family/FamilyHistory.docx
FILE://Documents/Letters/XmasLetter-2012.pdf
--toOo----OoOt---------
```

802b, 806b, 804b

FIG. 8B

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 92045

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

Motion Offense, LLC

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature	/Andrew Gordon/	Date (Optional)	
Name	Andrew Gordon		
Title	Manager, Motion Offense, LLC		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☐ Total of forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149D-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2:

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1 — Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Robert	Paul	Morris	

Residence Information (Select One) ● US Residency Non US Residency Active US Military Service

City	Madison	State/Province	GA	Country of Residence	US

Mailing Address of Inventor:

Address 1	751 Markhams Dr
Address 2	

City	Madison	State/Province	GA
Postal Code	30650	**Country**	US

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. — Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	92045		
Email Address	pcaldwell@thecaldwellfirm.com	Add Email	Remove Email

Application Information:

Title of the Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION		
Attorney Docket Number	DR0149D-CIP	**Small Entity Status Claimed**	☒
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)		**Suggested Figure for Publication (if any)**	

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149D-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☒ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	● Customer Number	US Patent Practitioner	◯ Limited Recognition (37 CFR 11.9)
Customer Number	92045		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.
When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
	Continuation of	15984406	2018-05-20

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149D-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15984406	Continuation of	15724223	2017-10-03	10021052	2018-07-10

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15724223	Continuation in part of	14274623	2014-05-09

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
14274623	Continuation in part of	13626635	2012-09-25

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
14274623	Continuation in part of	13624906	2012-09-22

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button. Add

Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)[i] the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Remove

Application Number	Country[i]	Filing Date (YYYY-MM-DD)	Access Code[i] (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button. Add

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149D-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149D-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☒ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149D-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Applicant 1 — Remove

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Clear

● Assignee	Legal Representative under 35 U.S.C. 117	Joint Inventor
Person to whom the inventor is obligated to assign.	Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor:

If the Applicant is an Organization check here. ☒

Organization Name	MOTION OFFENSE, LLC		
Mailing Address Information For Applicant:			
Address 1	211 W Tyler St., Suite C		
Address 2			
City	Longview	**State/Province**	TX
Country	US	Postal Code	75601
Phone Number		Fax Number	
Email Address			

Additional Applicant Data may be generated within this form by selecting the Add button. Add

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149D-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

Remove

If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1			
Address 2			
City		**State/Province**	
Country i		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button. Add

Signature:

Remove

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). **However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).**

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/Patrick Caldwell/			Date (YYYY-MM-DD)	2020-02-27
First Name	Patrick	Last Name	Caldwell	Registration Number	44580

Additional Signature may be generated within this form by selecting the Add button. Add

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DR0149D-CIP
	Application Number	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION	

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1 The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3 A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent CooperationTreaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

PTO/SB/08a (02-18)
Approved for use through 11/30/2020. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	
Filing Date	
First Named Inventor	Robert Paul Morris
Art Unit	
Examiner Name	
Attorney Docket Number	DR0149D-CIP

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	6192394	B1	2001-02-20	Gutfreund et al.	
	2	6502236	B1	2002-12-31	Allen et al.	
	3	6996599	B1	2006-02-07	Anders et al.	
	4	7051119	B2	2006-05-23	Shafron et al.	
	5	7080120	B2	2006-07-18	Betros et al.	
	6	7130883	B2	2006-10-31	Zhu et al.	
	7	7133897	B1	2006-11-07	Tran	
	8	7209953	B2	2007-04-24	Brooks	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	9	7328239	B1	2008-02-05	Berberian et al.	
	10	7409424	B2	2008-08-05	Parker	Entire Document
	11	7421069	B2	2008-09-02	Vernon et al.	
	12	7426578	B2	2008-09-16	Jones et al.	
	13	7529798	B2	2009-05-05	Rust	
	14	7533146	B1	2009-05-12	Kumar	
	15	7567553	B2	2009-07-28	Morris	
	16	7613772	B2	2009-11-03	Bartram et al.	
	17	7623650	B2	2009-11-24	Dalton et al.	
	18	7624148	B2	2009-11-24	Dalen	
	19	7653668	B1	2010-01-26	Shelat et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	20	7680950	B1	2010-03-16	Slaughter et al.	
	21	7701882	B2	2010-04-20	Jones et al.	
	22	7757265	B2	2010-07-13	Reynolds et al.	
	23	7765581	B1	2010-07-27	Caronni et al.	
	24	7783718	B2	2010-08-24	Modi	
	25	7809709	B1	2010-10-05	Shelton E. Harrison	
	26	7809854	B2	2010-10-05	Jung et al.	
	27	7904369	B1	2011-03-08	Andreasen et al.	
	28	7949722	B1	2011-05-24	Ullman et al.	
	29	7953853	B2	2011-05-31	Christensen et al.	
	30	8020190	B2	2011-09-13	Plummer	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	31	8060619	B1	2011-11-15	Saulpaugh et al.	
	32	8214329	B2	2012-07-03	Gilder et al.	
	33	8280948	B1	2012-10-02	Chen	
	34	8321499	B2	2012-11-27	Reisman	
	35	8332654	B2	2012-12-11	Anbuselvan	
	36	8359361	B2	2013-01-22	Thornton et al.	
	37	8583745	B2	2013-11-12	Bearman et al.	
	38	8671145	B2	2014-03-11	Roskowski et al.	
	39	8683576	B1	2014-03-25	Yun	
	40	8719251	B1	2014-05-06	English et al.	
	41	8775511	B2	2014-07-08	Vernon et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	42	8812733	B1	2014-08-19	Black et al.	
	43	8824643	B2	2014-09-02	Sahai	
	44	8825597	B1	2014-09-02	Houston et al.	Entire Document
	45	8849914	B2	2014-09-30	Bove et al.	
	46	8909710	B2	2014-12-09	Blackstock et al.	
	47	8996658	B2	2015-03-31	Anbuselvan	
	48	9053203	B2	2015-06-09	Wilson et al.	
	49	9299056	B2	2016-03-29	Kallman et al.	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20020087505	A1	2002-07-04	Smith et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	2	20020087522	A1	2002-07-04	MacGregor et al.	
	3	20020143855	A1	2002-10-03	Traversat et al.	
	4	20020147840	A1	2002-10-10	Mutton et al.	
	5	20030172127	A1	2003-09-11	Northrup et al.	
	6	20030200268	A1	2003-10-23	Morris	
	7	20040015610	A1	2004-01-22	Treadwell	
	8	20040024754	A1	2004-02-05	Mane et al.	
	9	20040044727	A1	2004-03-04	Abdelaziz et al.	
	10	20040064478	A1	2004-04-01	Canesi	
	11	20040064511	A1	2004-04-01	Abdel-Aziz et al.	
	12	20040088347	A1	2004-05-06	Yeager et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	13	20040117456	A1	2004-06-17	Brooks	
	14	20040158607	A1	2004-08-12	Coppinger et al.	Entire Document
	15	20040186894	A1	2004-09-23	Jhingan et al.	
	16	20050010646	A1	2005-01-13	Shiina	
	17	20050039130	A1	2005-02-17	Paul	Entire Document
	18	20050102348	A1	2005-05-12	Parsons et al.	
	19	20050114664	A1	2005-05-26	Davin	
	20	20050114671	A1	2005-05-26	Little et al.	
	21	20050223073	A1	2005-10-06	Malik	
	22	20060069787	A1	2006-03-30	Sinclair et al.	
	23	20060095531	A1	2006-05-04	Cho	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	24	20060224583	A1	2006-10-05	Fikes et al.	
	25	20060224608	A1	2006-10-05	Zamir et al.	
	26	20060224615	A1	2006-10-05	Korn et al.	
	27	20060224624	A1	2006-10-05	Korn et al.	
	28	20060224938	A1	2006-10-05	Fikes et al.	
	29	20060256934	A1	2006-11-16	Mazor	
	30	20070067726	A1	2007-03-22	Flynt et al.	
	31	20070078902	A1	2007-04-05	Buschi et al.	
	32	20070100713	A1	2007-05-03	Favero et al.	
	33	20070180150	A1	2007-08-02	Eisner et al.	
	34	20080005103	A1	2008-01-03	Ratcliffe et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	35	20080016160	A1	2008-01-17	Walter et al.	
	36	20080059500	A1	2008-03-06	Symens	
	37	20080098075	A1	2008-04-24	O'Bryan et al.	
	38	20080155110	A1	2008-06-26	Morris	
	39	20080201364	A1	2008-08-21	Buschi et al.	
	40	20080256199	A1	2008-10-16	Pesala	
	41	20080320094	A1	2008-12-25	Tu et al.	
	42	20090157829	A1	2009-06-18	Choi et al.	
	43	20090157859	A1	2009-06-18	Morris	
	44	20090177526	A1	2009-07-09	Aaltonen et al.	
	45	20090222450	A1	2009-09-03	Zigelman	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	46	20090254601	A1	2009-10-08	Moeller et al.	
	47	20090265426	A1	2009-10-22	Svendsen et al.	
	48	20090327904	A1	2009-12-31	Guzak et al.	
	49	20100011077	A1	2010-01-14	Shkolnikov et al.	
	50	20100070602	A1	2010-03-18	Dale W. Malik	
	51	20100169442	A1	2010-07-01	Liu et al.	
	52	20100235439	A1	2010-09-16	Goodnow	
	53	20100312782	A1	2010-12-09	Li et al.	
	54	20110072033	A1	2011-03-24	White et al.	
	55	20110138004	A1	2011-06-09	McConn et al.	
	56	20110276637	A1	2011-11-10	Thornton et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	57	20120011207	A1	2012-01-12	Morris	
	58	20120011444	A1	2012-01-12	Morris	
	59	20130046992	A1	2013-02-21	Resch et al.	
	60	20130283189	A1	2013-10-24	Basso et al.	Entire Document
	61	20140067929	A1	2014-03-06	Kirigin et al.	Entire Document
	62	20140089419	A1	2014-03-27	Morris	
	63	20140089420	A1	2014-03-27	Morris	
	64	20140089421	A1	2014-03-27	Morris	
	65	20140108487	A1	2014-04-17	Rosset et al.	
	66	20140112319	A1	2014-04-24	Morris	
	67	20140172999	A1	2014-06-19	Morris	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	68	20140279050	A1	2014-09-18	Makar et al.		
	69	20150331571	A1	2015-11-19	Chaudhri	Entire Document	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1	2002239940	AU	A1	2002-07-30	Paul et al.	Entire Document	
	2	2002077773	WO	A3	2009-06-11	Ross		

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	Berners-Lee, T, et al, "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, pages 1-61, 2005-01, Internet Engineering Task Force (IEFT), http://www.ietf.org/rfc/rfc3986.txt	
	2	Blog Post, "US Postal Service "Real Mail Notification" Comes Thru!," Sky's Blog, July 31, 2012, available at https://blog.red7.com/real-mail-notification/	
	3	Crocker, D, "Standard for the Format of ARPA Internet Text Messages", RFC 822, pages 1-47, 1982-08, Internet Engineering Task Force (IEFT), http://www.ietf.org/rfc/rfc822.txt	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	4	Fielding, R, et al, "Hypertext Transfer Protocol -- HTIP/1.1", RFC 2616, pages 1-114, 1999-06, Internet Engineering Task Force (IEFT), http://www.ietf.org/rfc/rfc2616.txt	
	5	FREED, N, et al, "Multipurpose Internet Mail Extensions (MIME) Part 1: Format of Internet Message Bodies", RFC 2045, pages 1-31, 1996-11, Internet Engineering Task Force (IEFT), http://www.ietf.org/rfc/rfc2045.txt	
	6	Hoffman, P, et al, "The mailto URL scheme", RFC 2368, pages 1-10, 1998-07, Internet Engineering Task Force (IE FT), http :/fwww.ietf.org/rfc/rfc2368. txt	
	7	Instructional & Research Information Systems (IRIS), "Outlook LDAP Autocomplete," Page published August 7, 2007, available at https://iris.eecs.berkeley.edu/faq/software/outlook/outlook-ldap-autocomplete/	
	8	Klensin, J, "Simple Mail Transfer Protocol", RFC 5321, pages 1-95, 2008-10, Internet Engineering Task Force (IEFT), http://www.ietf.org/rfc/rfc5321.txt	
	9	Myers, J, et al, "Post Office Protocol - version 3", RFC 1939, pages 1-23, 1996-05, Internet Engineering Task Force (IEFT), http://www.ietf.org/rfc/rfc1939.txt	
	10	Notice of Allowance in U.S. Pat. Appl. No. 14/924/677 dated 06-13-2016	
	11	Office Action Summary in U.S. Pat. App. No. 12819215 dated 02262014	
	12	Office Action Summary in U.S. Pat. App. No. 12819215 dated 03022015	
	13	Office Action Summary in U.S. Pat. App. No. 12819215 dated 03282013	
	14	Office Action Summary in U.S. Pat. App. No. 12833014 dated 08022012	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	15	Office Action Summary in U.S. Pat. App. No. 12833016 dated 02142014	
	16	Office Action Summary in U.S. Pat. App. No. 12833016 dated 05102013	
	17	Office Action Summary in U.S. Pat. App. No. 12833016 dated 09202012	
	18	Office Action Summary in U.S. Pat. App. No. 13624906 dated 12312014	
	19	Office Action Summary in U.S. Pat. App. No. 13624940 dated 07182014	
	20	Office Action Summary in U.S. Pat. App. No. 13626635 dated 08292014	
	21	Office Action Summary in U.S. Pat. App. No. 13647144 dated 12012014	
	22	Office Action Summary in U.S. Pat. App. No. 13654647 dated 09252014	
	23	Office Action Summary in U.S. Pat. App. No. 13716156 dated 03262015	
	24	Office Action Summary in U.S. Pat. App. No. 13716158 dated 05082015	
	25	Office Action Summary in U.S. Pat. App. No. 13716159 dated 10072015	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	26	Office Action Summary in U.S. Pat. App. No. 13716160 dated 10262015	
	27	Office Action Summary in U.S. Pat. App. No. 14274623 dated 03292017	
	28	Office Action Summary in U.S. Pat. App. No. 14274623 dated 10032016	
	29	Office Action Summary in U.S. Pat. App. No. 15934909 dated 06052018	
	30	RESNICK, E, "Internet Message Format", RFC 5322, pages 1-57, 2008-10, Internet Engineering Task Force (IEFT), http://www.ietf.org/rfc/rfc5322.txt	
	31	RICHARDSON T, et al, "The Remote Framebuffer Protocol", RFC 6143, pages 1-39, 2011-03, Internet Engineering Task Force (IEFT), http://www.ietf.org/rfc/rfc6143.txt	
	32	Rosenberg, J, et al, "SIP: Session Initiation Protocol", RFC 3261, pages 1-269, 2002-06, Internet Engineering Task Force (IEFT), http://www.ietf.org/rfc/rfc3261.txt	
	33	USPS, "Postal officials expanding services at Stewart Station Post Office," July 24, 2012, available at https://about.usps.com/news/state-releases/ny/2012/ny_2012_0724.htm	
	34	Web Archive Webpage, "About ifttt," Page dated Jan 6, 2012, available at https://web.archive.org/web/20120106190055/http:/ifttt.com/wtf	
	35	Web Archive Webpage, "Attachmore How it works - The basics," page dated June 9, 2012, available at https://web.archive.org/web/20120609073922/http:/www.attachmore.com/Learn/HowItWorks.aspx	
	36	Web Archive Webpage, "Attachmore Tour," webpage date June 6, 2012, available at https://web.archive.org/web/20120606140221/http:/www.attachmore.com/Learn/Tour.aspx#tabs	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

	37	Web Archive Webpage, "How do I link to files in my Dropbox," page dated Jan 7, 2012, available at https://web.archive.org/web/20120107115329/http:/www.dropbox.com/help/16
	38	Wilde, E, et al, "URI Scheme for Global System for Mobile Communications (GSM) Short Message Service", RFC 5724, pages 1-18, 2010-01, Internet Engineering Task Force (IEFT), http://www.ietf.org/rfc/rfc5724.txt

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	Robert Paul Morris
	Art Unit	
	Examiner Name	
	Attorney Docket Number	DR0149D-CIP

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Patrick Caldwell/	Date (YYYY-MM-DD)	2020-02-27
Name/Print	Patrick Caldwell	Registration Number	44580

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION

RELATED APPLICATIONS

[0001] The present application is a continuation of and claims priority to U.S. Patent Application No.15/984,406 filed 05-20-2018 and entitled “METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION,” which is a continuation of and claims priority to U.S. Patent Application No. 15/724,223 filed 10-03-2017 and entitled “METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION” which is a continuation-in-part of and claims priority to U.S. Patent Application No. 14/274,623 (published US 2014-0365588 A1) filed 05/09/2014 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION," which, in turn, is a continuation-in-part of and claims priority to U.S. Patent Application No. 13/624,906 (published US 2014-0089419 A1) filed 09/22/2012 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION" and U.S. Patent Application No. 13/626,635 (published under US 2014-0089421 A1) filed 09/25/2012 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR SHARING A DATA OBJECT IN A DATA STORE VIA A COMMUNICATION."

[0002] This application is related to the following U.S. Patent Applications:

[0003] U.S. Patent Application No. 12/833,016 (published US 2012-0011444 A1) filed on 2010/07/09, entitled “METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR REFERENCING AN ATTACHMENT IN A COMMUNICATION”;

[0004] U.S. Patent Application No. 12/833,014 (published US 2012-0011207 A1) filed on 2010/07/09, entitled “METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A REQUEST FOR A RESOURCE IN A COMMUNICATION”;

[0005] U.S. Patent Application No. 13/624,906 (published US 2014-0089419 A1) filed 2012/09/22 and entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION;" AND

[0006] U.S. Patent Application No. 13/626,635 (published under US 2014-0089421 A1) filed 2012/09/25 entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR SHARING A DATA OBJECT IN A DATA STORE VIA A COMMUNICATION;" the entire contents of each are herein incorporated by reference for all purposes.

BACKGROUND

[0007] While receiving attachments in a network communication, such as email, is common, most attachments received are unsolicited or unrequested. To request an attachment, a user typically sends the request as voice and/or text data in a voice and/or text message heard and/or read by another user. The request may be as vague or as specific as the language used by the requesting user. The other user must interpret the request and find a file or other data object that seems to match the request. The other user in many cases locates a “matching” resource using a program or application other than the communications agent that received the request. For example, a search program may be used to search a hard-drive by the other user. The other user must enter the search criteria.

[0008] Accordingly, there exists a need for methods, systems, and computer program products for processing a data object identification request in a communication. .

SUMMARY

[0009] The following presents a simplified summary of the disclosure in order to provide a basic understanding to the reader. This summary is not an extensive overview of the disclosure and it does not identify key/critical elements of the invention or delineate the scope of the invention. Its sole purpose is to present some concepts disclosed herein in a simplified form as a prelude to the more detailed description that is presented later.

[0010] In one embodiment, an apparatus is provided, comprising: at least one non-transitory memory storing instructions; and one or more processors in communication with the at least one non-transitory memory, wherein the one or more processors execute the instructions to: cause, at a first node, display of at least one first interface with a first user interface element and a second user interface element, utilizing first hypertext markup language-equipped code that is sent via at least one network; receive, from the first node via the at least one network, a file description via the first user interface element that includes a textbox, utilizing the at least one first interface; receive, from the first node via the at least one network, an indication of at least one of a date or a time via the second user interface element, utilizing the at least one first interface; cause, at the first node, display of at least one second interface with a third user interface element and a fourth user interface element, utilizing second hypertext markup language-equipped code that is sent via the at least one network; receive, from the first node via the at least one network, indicia associated with at least one email address via the third user interface element, utilizing the at least one second interface; receive, from the first node via the at least one network, an indication of a selection of the fourth user interface element, utilizing the at least one second interface; in response to the receipt of the indication of the selection of the fourth user interface element and based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; generate a first message indicating the at least one of the date or the time, and including a fifth user interface element and at least one sentence including the file description; send, to a second node via the at least one network, the first message; receive, from the second node via the at least one

network, an indication of a selection of the fifth user interface element of the first message; in response to the receipt of the indication of the selection of the fifth user interface element, cause, at the second node, display of at least one third interface with a sixth user interface element and a seventh user interface element, utilizing third hypertext markup language-equipped code that is sent via at least one network; receive, from the second node via the at least one network, an indication of a selection at least one file utilizing the sixth user interface element, utilizing the at least one third interface; receive, from the second node via the at least one network, an indication of a selection of the seventh user interface element, utilizing the at least one third interface; in response to the receipt of the indication of the selection of the seventh user interface element, generate a second message that includes an eighth user interface element, and that does not include a file attachment with the second message; send, to the first node via the at least one network, the second message; receive, from the first node via the at least one network, an indication of a selection of the eighth user interface element of the second message; in response to the receipt of the indication of a selection of the eighth user interface element of the second message, cause, at the first node, display of at least one fourth interface with a reference to the at least one file for allowing access, at the first node, to the at least one file.

[0011] In another embodiment, an apparatus is provided, comprising at least one non-transitory memory storing instructions; and one or more processors in communication with the at least one non-transitory memory, wherein the one or more processors execute the instructions to: send a first communication over at least one network that results in display, at a first node, of a first user interface element, a second user interface element, and a third user interface element; receive, from the first node via the at least one network, text associated with one or more desired files via the first user interface element that includes a textbox; receive, from the first node via the at least one network, an object associated with at least one email address via the second user interface element; receive, from the first node via the at least one network, a message associated with a selection of the third user interface element; in response to the receipt of the message associated with the selection of the third user interface element and based on the text associated with the one or more desired files and the object

associated with the at least one email address; generate a first message including a fourth user interface element and the text associated with the one or more desired files; send, to a second node via the at least one network, the first message; receive, from the second node via the at least one network, a message associated with a selection of the fourth user interface element of the first message; in response to the receipt of the message associated with the selection of the fourth user interface element, send a third communication over the at least one network that results in display, at the second node, of a fifth user interface element; receive, from the second node via the at least one network, a message associated with a selection at least one file; receive, from the second node via the at least one network, a message associated with a selection of the fifth user interface element; in response to the receipt of the message associated with the selection of the fifth user interface element, generate a second message that includes an sixth user interface element, and that does not include a file attachment with the second message; send, to the first node via the at least one network, the second message; receive, from the first node via the at least one network, a message associated with a selection of the sixth user interface element of the second message; in response to the receipt of the message associated with the selection of the sixth user interface element of the second message, send a fourth communication over the at least one network that results in display, at the first node, of a reference to the at least one file; and allow access, at the first node, to the at least one file via the reference.

[0012] In yet another embodiment, a computer-implemented method is provided, comprising: displaying, at a first node, at least one first interface with a first user interface element and a second user interface element, utilizing first hypertext markup language-equipped code that is received via at least one network; receiving, utilizing the at least one first interface at the first node, a file description via the first user interface element that includes a textbox; sending, from the first node via the at least one network, the file description; receiving, utilizing the at least one first interface at the first node, an indication of at least one of a date or a time via the second user interface element; sending, from the first node via the at least one network, the indication of the at least one of the date or the time; displaying, at the first node, at least one second interface with a third user interface element and a fourth user interface element, utilizing

second hypertext markup language-equipped code that is received via the at least one network; receiving, utilizing the at least one second interface at the first node, indicia associated with at least one email address via the third user interface element; sending, from the first node via the at least one network, the indicia associated with the at least one email address; detecting, at the first node, a selection of the fourth user interface element; sending, from the first node via the at least one network, an indication of the selection of the fourth user interface element; after sending the indication of the selection of the fourth user interface element and based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; receiving, at a second node via the at least one network, a first message indicating the at least one of the date or the time, and including a fifth user interface element and at least one sentence including the file description; detecting, at the second node, a selection of the fifth user interface element of the first message; sending, from the second node via the at least one network, an indication of the selection of the fifth user interface element of the first message; after sending the indication of the selection of the fifth user interface element, displaying, at the second node, at least one third interface with a sixth user interface element and a seventh user interface element, utilizing third hypertext markup language-equipped code that is received via at least one network; receiving, utilizing the at least one third interface at the second node, a selection of at least one file utilizing the sixth user interface element; sending, from the second node via the at least one network, an indication of the at least one file; detecting, utilizing the at least one third interface at the second node, a selection of the seventh user interface element; sending, from the second node via the at least one network, an indication of the selection of the seventh user interface element; after sending the indication of the selection of the seventh user interface element, receiving, at the first node, a second message that includes an eighth user interface element, and that does not include a file attachment with the second message; detecting, at the first node, a selection of the eighth user interface element of the second message; sending, from the first node via the at least one network, an indication of the selection of the eighth user interface element of the second message; after sending the indication of a selection of the eighth user interface element of the second

message, displaying, at the first node, at least one fourth interface with a reference to the at least one file; detecting, utilizing the at least one fourth interface at the first node, a selection of the reference to the at least one file; and providing access, at the first node, to the at least one file, utilizing the reference.

[0013] In still yet another embodiment, an apparatus is provided, comprising: a first node configured to: display, at the first node, a first user interface element, a second user interface element, and a third user interface element; receive, at the first node, text associated with one or more desired files via the first user interface element that includes a textbox; send, from the first node via at least one network, the text associated with the one or more desired files; receive, at the first node, an object associated with at least one email address via the second user interface element; send, from the first node via the at least one network, the object associated with the at least one email address; detect, at the first node, a selection of the third user interface element; and send, from the first node via the at least one network, a message associated with the selection of the third user interface element; said first node further configured to communicate with a second node that is configured to: after the message associated with the selection of the third user interface element is sent and based on the text associated with the one or more desired files and the object associated with the at least one email address; receive, at the second node via the at least one network, a first message including a fourth user interface element and at least one sentence including the text associated with the one or more desired files; detect, at the second node, a selection of the fourth user interface element of the first message; send, from the second node via the at least one network, a message associated with the selection of the fourth user interface element of the first message; after the message associated with the selection of the fourth user interface element is sent, display, at the second node, a fifth user interface element and a sixth user interface element; detect, at the second node, a selection of at least one file; send, from the second node via the at least one network, a message associated with the at least one file; detect, at the second node, a selection of the sixth user interface element; and send, from the second node via the at least one network, a message associated with the selection of the sixth user interface element; said first node further configured to: after the message associated

with the selection of the sixth user interface element is sent, receive, at the first node, a second message that includes a seventh user interface element, and that does not include a file attachment with the second message; detect, at the first node, a selection of the seventh user interface element of the second message; send, from the first node via the at least one network, a message associated with the selection of the seventh user interface element of the second message; after the message associated with the selection of the seventh user interface element of the second message is sent, display, at the first node, an eighth user interface element associated with a reference to the at least one file; and provide access, at the first node, to the at least one file utilizing the reference.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] Objects and advantages of the present invention will become apparent to those skilled in the art upon reading this description in conjunction with the accompanying drawings, in which like reference numerals have been used to designate like or analogous elements, and in which:

[0015] FIG. **1** is a block diagram illustrating an exemplary hardware device included in and/or otherwise providing an execution environment in which the subject matter may be implemented;

[0016] FIG. **2A** is a flow diagram illustrating a method for processing a data object identification request in a communication according to an aspect of the subject matter described herein;

[0017] FIG. **2B** is a flow diagram illustrating a method for processing a data object identification request in a communication according to an aspect of the subject matter described herein;

[0018] FIG. **3A** is a block diagram illustrating an arrangement of components for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0019] FIG. **3B** is a block diagram illustrating an arrangement of components for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0020] FIG. **4** is a block diagram illustrating an arrangement of components for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0021] FIG. **5** is a network diagram illustrating a system for processing a data object identification request in a communication according to another aspect of the subject matter described herein;

[0022] FIG. **6A** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0023] FIG. **6B** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0024] FIG. **6C** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0025] FIG. **6D** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0026] FIG. **6E** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0027] FIG. **6F** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0028] FIG. **6G** is a diagram illustrating a user interface presented via a display according to another aspect of the subject matter described herein;

[0029] FIG. **7** is a message flow diagram illustrating an exemplary data and execution flow for processing a data object identification request in a communication according to an aspect of the subject matter described herein;

[0030] FIG. **8A** illustrates an exemplary portion of a communication between a first execution environment and a second execution environment according to an aspect of the subject matter described herein; and

[0031] FIG. **8B** illustrates another exemplary portion of a communication between a first execution environment and a second execution environment according to an aspect of the subject matter described herein.

DETAILED DESCRIPTION

[0032] One or more aspects of the disclosure are described with reference to the drawings, wherein like reference numerals are generally utilized to refer to like elements throughout, and wherein the various structures are not necessarily drawn to scale. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of one or more aspects of the disclosure. It may be evident, however, to one skilled in the art, that one or more aspects of the disclosure may be practiced with a lesser degree of these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate describing one or more aspects of the disclosure.

[0033] Unless otherwise defined, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this disclosure belongs. Although methods, components, and devices similar or equivalent to those described herein can be used in the practice or testing of the subject matter described herein, suitable methods, components, and devices are described below.

[0034] All publications, patent applications, patents, and other references mentioned herein are incorporated by reference in their entirety. In case of conflict, the present specification, including definitions, will control. In addition, the materials, methods, and examples are illustrative only and not intended to be limiting.

[0035] An exemplary device included in an execution environment that may be configured according to the subject matter is illustrated in FIG. **1**. An "execution environment", as used herein, is an arrangement of hardware and, in some aspects, software that may be further configured to include and/or otherwise host an arrangement of components for performing a method of the subject matter described herein. An execution environment includes and/or is otherwise provided by one or more devices. The execution environment is said to be the execution environment of the device and/or devices. An execution environment may be and/or may include a virtual execution environment including software components operating in a host execution environment. Exemplary devices included in and/or otherwise providing suitable execution environments for configuring according to the subject matter include personal computers, notebook computers, tablet computers, servers, portable electronic devices, handheld electronic devices, mobile devices, multiprocessor devices, distributed systems, consumer electronic devices, routers, communication servers, and/or any other suitable devices. Those skilled in the art will understand that the components illustrated in FIG. **1** are exemplary and may vary by particular execution environment.

[0036] FIG. **1** illustrates a hardware device **100** included in an execution environment **102**. FIG. **1** illustrates that execution environment **102** includes a processor **104**, such as one or more microprocessors; a physical processor memory **106** including storage locations identified by addresses in a physical memory address space of processor **104**; a persistent secondary storage **108**, such as one or more hard drives and/or flash

storage media; an input device adapter **110**, such as a key or keypad hardware, a keyboard adapter, and/or a mouse adapter; an output device adapter **112**, such as a display and/or an audio adapter for presenting information to a user; a network interface component, illustrated by a network interface adapter **114**, for communicating via a network such as a LAN and/or WAN; and a communication mechanism that operatively couples elements **104-114**, illustrated as a bus **116**. Elements **104-114** may be operatively coupled by various means. Bus **116** may comprise any type of bus architecture, including a memory bus, a peripheral bus, a local bus, and/or a switching fabric.

[0037] As used herein a "processor" is an instruction execution machine, apparatus, or device. A processor may include one or more electrical, optical, and/or mechanical components that operate in interpreting and executing program instructions. Exemplary processors include one or more microprocessors, digital signal processors (DSPs), graphics processing units, application-specific integrated circuits (ASICs), optical or photonic processors, and/or field programmable gate arrays (FPGAs). Processor **104** may access machine code instructions and data via one or more memory address spaces in addition to the physical memory address space. A memory address space includes addresses identifying locations in a processor memory. The addresses in a memory address space are included in defining a processor memory. Processor **104** may have more than one processor memory. Thus, processor **104** may have more than one memory address space. Processor **104** may access a location in a processor memory by processing an address identifying the location. The processed address may be identified by an operand of a machine code instruction and/or may be identified by a register or other portion of processor **104**.

[0038] FIG. **1** illustrates a virtual processor memory **118** spanning at least part of physical processor memory **106** and may span at least part of persistent secondary storage **108**. Virtual memory addresses in a memory address space may be mapped to physical memory addresses identifying locations in physical processor memory **106**. An address space for identifying locations in a virtual processor memory is referred to as a virtual memory address space; its addresses are referred to as virtual memory addresses; and its processor memory is referred to as a virtual processor memory or

virtual memory. The term "processor memory" may refer to physical processor memory, such as processor memory **106**, and/or may refer to virtual processor memory, such as virtual processor memory **118**, depending on the context in which the term is used.

[0039] Physical processor memory **106** may include various types of memory technologies. Exemplary memory technologies include static random access memory (SRAM) and/or dynamic RAM (DRAM) including variants such as dual data rate synchronous DRAM (DDR SDRAM), error correcting code synchronous DRAM (ECC SDRAM), RAMBUS DRAM (RDRAM), and/or XDR™ DRAM. Physical processor memory **106** may include volatile memory as illustrated in the previous sentence and/or may include nonvolatile memory such as nonvolatile flash RAM (NVRAM) and/or ROM.

[0040] Persistent secondary storage **108** may include one or more flash memory storage devices, one or more hard disk drives, one or more magnetic disk drives, and/or one or more optical disk drives. Persistent secondary storage may include a removable data storage medium. The drives and their associated tangible computer readable storage media provide volatile and/or nonvolatile storage for computer-readable instructions, data structures, program components, and other data for execution environment **102**.

[0041] Execution environment **102** may include software components stored in persistent secondary storage **108**, in remote storage accessible via a network, and/or in a processor memory. FIG. **1** illustrates execution environment **102** including an operating system **120**, one or more applications **122**, and other program code and/or data components illustrated by other libraries and subsystems **124**. In an aspect, some or all software components may be stored in locations accessible to processor **104** in a shared memory address space shared by the software components. The software components accessed via the shared memory address space are stored in a shared processor memory defined by the shared memory address space. In another aspect, a first software component may be stored in one or more locations accessed by processor **104** in a first address space and a second software component may be stored in one or more locations accessed by processor **104** in a second address space. The first software component is stored in a first processor memory defined by the first address

space and the second software component is stored in a second processor memory defined by the second address space.

[0042] Software components typically include instructions executed by processor **104** in a computing context referred to as a "process". A process may include one or more "threads". A "thread" includes a sequence of instructions executed by processor **104** in a computing sub-context of a process. The terms "thread" and "process" may be used interchangeably herein when a process includes only one thread.

[0043] Execution environment **102** may receive user-provided information via one or more input devices illustrated by an input device **128**. Input device **128** provides input information to other components in execution environment **102** via input device adapter **110**. Execution environment **102** may include an input device adapter for a keyboard, a touch screen, a microphone, a joystick, a television receiver, a video camera, a still camera, a document scanner, a fax, a phone, a modem, a network interface adapter, and/or a pointing device, to name a few exemplary input devices.

[0044] Input device **128** included in execution environment **102** may be included in device **100** as FIG. **1** illustrates or may be external (not shown) to device **100**. Execution environment **102** may include one or more internal and/or external input devices. External input devices may be connected to device **100** via corresponding communication interfaces such as a serial port, a parallel port, and/or a universal serial bus (USB) port. Input device adapter **110** receives input and provides a representation to bus **116** to be received by processor **104**, physical processor memory **106**, and/or other components included in execution environment **102**.

[0045] An output device **130** in FIG. **1** exemplifies one or more output devices that may be included in and/or that may be external to and operatively coupled to device **100**. For example, output device **130** is illustrated connected to bus **116** via output device adapter **112**. Output device **130** may be a display device. Exemplary display devices include liquid crystal displays (LCDs), light emitting diode (LED) displays, and projectors. Output device **130** presents output of execution environment **102** to one or more users. In some embodiments, an input device may also include an output device. Examples include a phone, a joystick, and/or a touch screen. In addition to various types of display devices, exemplary output devices include printers, speakers, tactile

output devices such as motion-producing devices, and other output devices producing sensory information detectable by a user. Sensory information detected by a user is referred herein to as "sensory input" with respect to the user.

[0046] A device included in and/or otherwise providing an execution environment may operate in a networked environment communicating with one or more devices via one or more network interface components. The terms "communication interface component" and "network interface component" are used interchangeably herein. FIG. **1** illustrates network interface adapter (NIA) **114** as a network interface component included in execution environment **102** to operatively couple device **100** to a network. A network interface component includes a network interface hardware (NIH) component and optionally a network interface software (NIS) component.

[0047] Exemplary network interface components include network interface controller components, network interface cards, network interface adapters, and line cards. A node may include one or more network interface components to interoperate with a wired network and/or a wireless network. Exemplary wireless networks include a BLUETOOTH network, a wireless 802.11 network, and/or a wireless telephony network (e.g., a cellular, PCS, CDMA, and/or GSM network). Exemplary network interface components for wired networks include Ethernet adapters, Token-ring adapters, FDDI adapters, asynchronous transfer mode (ATM) adapters, and modems of various types. Exemplary wired and/or wireless networks include various types of LANs, WANs, and/or personal area networks (PANs). Exemplary networks also include intranets and internets such as the Internet.

[0048] The terms "network node" and "node" in this document both refer to a device having a network interface component for operatively coupling the device to a network. Further, the terms "device" and "node" used herein may refer to one or more devices and nodes, respectively, providing and/or otherwise included in an execution environment unless clearly indicated otherwise.

[0049] The user-detectable outputs of a user interface are generically referred to herein as "user interface elements" or abbreviated as "UI elements". More specifically, visual outputs of a user interface are referred to herein as "visual interface elements". A visual interface element may be a visual output of a graphical user interface (GUI).

Exemplary visual interface elements include windows, textboxes, sliders, list boxes, drop-down lists, spinners, various types of menus, toolbars, ribbons, combo boxes, tree views, grid views, navigation tabs, scrollbars, labels, tooltips, text in various fonts, balloons, dialog boxes, and various types of button controls including check boxes and radio buttons. An application interface may include one or more of the elements listed. Those skilled in the art will understand that this list is not exhaustive. The terms “visual representation”, “visual output”, and “visual interface element” are used interchangeably in this document. Other types of UI elements include audio outputs referred to as ”audio interface elements”, tactile outputs referred to as “tactile interface elements”, and the like.

[0050] A visual output may be presented in a two-dimensional presentation where a location may be defined in a two-dimensional space having a vertical dimension and a horizontal dimension. A location in a horizontal dimension may be referenced according to an X-axis and a location in a vertical dimension may be referenced according to a Y-axis. In another aspect, a visual output may be presented in a three-dimensional presentation where a location may be defined in a three-dimensional space having a depth dimension in addition to a vertical dimension and a horizontal dimension. A location in a depth dimension may be identified according to a Z-axis. A visual output in a two-dimensional presentation may be presented as if a depth dimension existed allowing the visual output to overlie and/or underlie some or all of another visual output.

[0051] An order of visual outputs in a depth dimension is herein referred to as a "Z-order". The term “Z-value” as used herein refers to a location in a Z-order. A Z-order specifies the front-to-back and/or back-to-front ordering of visual outputs in a presentation space with respect to a Z-axis. In one aspect, a visual output with a higher Z-value than another visual output may be defined to be on top of or closer to the front than the other visual output. In another aspect, a visual output with a lower Z-value than another visual output may be defined to be on top of or closer to the front than the other visual output.

[0052] A “user interface (UI) element handler” component, as the term is used herein, includes a component of configured to send information representing a program entity for presenting a user-detectable representation of the program entity by an output

device, such as a display. A “program entity” is an object included in and/or otherwise processed by an application or executable. The user-detectable representation is presented based on the sent information. Information that represents a program entity for presenting a user detectable representation of the program entity by an output device is referred to herein as “presentation information”. Presentation information may include and/or may otherwise identify data in one or more formats. Exemplary formats include image formats such as raw pixel data, JPEG, video formats such as MP4, markup language data such as hypertext markup language (HTML) and other XML-based markup, a bit map, and/or instructions such as those defined by various script languages, byte code, and/or machine code. For example, a web page received by a user agent from a remote application provider may include HTML, ECMAScript, and/or byte code for presenting one or more UI elements included in a user interface of the remote application. Components configured to send information representing one or more program entities for presenting particular types of output by particular types of output devices include visual interface element handler components, audio interface element handler components, tactile interface element handler components, and the like.

[0053] A representation of a program entity may be stored and/or otherwise maintained in a presentation space. As used in this document, the term “presentation space” refers to a storage region allocated and/or otherwise provided for storing presentation information, which may include audio, visual, tactile, and/or other sensory data for presentation by and/or on an output device. For example, a buffer for storing an image and/or text string may be a presentation space as sensory information for a user. A presentation space may be physically and/or logically contiguous or non-contiguous. A presentation space may have a virtual as well as a physical representation. A presentation space may include a storage location in a processor memory, secondary storage, a memory of an output adapter device, and/or a storage medium of an output device. A screen of a display, for example, is a presentation space.

[0054] As used herein, the terms "program” and "executable” refer to any data representation that may be translated into a set of machine code instructions and may optionally include associated program data. The terms are used interchangeably herein.

Program representations other than machine code include object code, byte code, and source code. Object code includes a set of instructions and/or data elements that either are prepared for linking prior to loading or are loaded into an execution environment. When in an execution environment, object code may include references resolved by a linker and/or may include one or more unresolved references. The context in which this term is used will make clear the state of the object code when it is relevant. This definition can include machine code and virtual machine code, such as Java™ byte code. As used herein, the terms “application”, and “service” may be realized in one or more executables and/or in one or more hardware components. The terms are used interchangeably herein.

[0055] As used herein, an “addressable entity” is program or a portion thereof, specifiable in programming language in source code. An addressable entity is addressable in a program component translated for a compatible execution environment from the source code. Examples of addressable entities include variables, constants, functions, subroutines, procedures, modules, methods, classes, objects, code blocks, and labeled instructions. A code block includes one or more instructions in a given scope specified in a programming language. An addressable entity may include a value. In some places in this document “addressable entity” refers to a value of an addressable entity. In these cases, the context will clearly indicate that the value is being referenced.

[0056] Addressable entities may be written in and/or translated to a number of different programming languages and/or representation languages, respectively. An addressable entity may be specified in and/or translated into source code, object code, machine code, byte code, and/or any intermediate languages for processing by an interpreter, compiler, linker, loader, and/or other analogous tool.

[0057] As used herein, the term “network protocol” refers to a formal set of rules, conventions and data structures that governs how computers and other network devices exchange information over a network. The rules, conventions, and data structures are said to be specified or defined in a specification and/or schema.

[0058] The term "network path" as used herein refers to a sequence of nodes in a network that are communicatively coupled for transmitting data in one or more data units of a specified network protocol between a pair of nodes in the network.

[0059] A "data unit", as the term is used herein, is a data entity specified according to a network protocol for transmitting data contained in and/or otherwise identified by the data entity. A data unit is transmitted between a pair of nodes in a network path to send the data in and/or otherwise identified by the data unit from a source node to a destination node that includes a protocol endpoint of the network protocol. A network protocol explicitly and/or implicitly specifies and/or otherwise identifies a schema that defines one or more of a format for a valid data unit and a vocabulary for content of a valid data unit. One example of a data unit is an Internet Protocol (IP) packet. The Internet Protocol defines a format for an IP packet that defines a header for identifying a destination address that identifies a destination node and a payload portion for including a representation of data to be delivered to the identified destination node. The terms "data unit", "data packet", and "packet" are used interchangeably herein. One or more data units of a first network protocol may transmit a message of second network protocol. For example, one or more data units of the IP protocol may include a TCP message. In another example, one or more TCP data units may transmit an HTTP message.

[0060] How data is packaged in one more data units for a network protocol may vary as the data traverses a network path from a source node to a destination node. Data may be transmitted in a single data unit between two consecutive nodes in a network path. Additionally, data may be exchanged between a pair of consecutive nodes in several data units each including a portion of the data. Data received in a single data unit by a node in a network path may be split into portions included in several respective data units for transmitting to a next node in the network path. Portions of data received in several data units may be combined into a single data unit for transmitting by a node in a network path. For purposes of describing the subject matter, a data unit in which data is received by a node is referred to as a different data unit than a data unit in which the data is forwarded by the node.

[0061] A “protocol address”, as the term is used herein, for a network protocol is an identifier of a protocol endpoint that may be represented in a data unit of the protocol. For example 192.168.1.1 is an IP protocol address represented in a human readable format that may be included in an address portion of an IP header to identify a source and/or a destination IP protocol endpoint. A protocol address differs from a symbolic identifier, defined below, in that a symbolic identifier, with respect to a network protocol, maps to a protocol address. Thus, “www. mynode.com” may be a symbolic identifier for a node in a network when mapped to the protocol address 192.168.1.1. An identifier may be both a symbolic identifier and a protocol address depending on its role with respect to its use for a particular network protocol.

[0062] Since a protocol endpoint is included in a node and is accessible via a network via a network interface, a protocol address identifies a node and identifies a network interface of the node. A network interface may include one or more NICs operatively coupled to a network.

[0063] A node in a pair of nodes in a network path at one end of the sequence of nodes in the network path and/or the other end is referred to herein as a “path end node”. Note that a node may have two NICs with one NIC at each end of a network path. A network path may be included as a portion of another network path that communicatively couples a same pair of nodes. Data may be transmitted via the sequence of nodes in a network path between path end nodes communicatively coupled via the network path. Data may be transmitted in one or both directions depending on an ordering of the nodes in the sequence.

[0064] The term “hop” as used herein refers to a pair of consecutive nodes in a network path for transmitting, via a network protocol, data sent from a source node to a destination node. A “hop path” is thus a sequence of hops in a network that respectively include a sequence of pairs of consecutive nodes included in transmitting data from a first path end node of the network path to a second path end node of the network path.

[0065] The term “path-based protocol address” as used herein refers to a protocol address for a network protocol that includes a plurality of path segment identifiers that identify portions of a specific network path identified by the path-based protocol address. A “node-based protocol address” is a path-based protocol address that

includes a plurality of node identifiers that identify a sequence of nodes in a network path. A "network-interface-based protocol address" is a path-based protocol address that includes a plurality of network interface identifiers that identify a sequence of network interfaces in a network path. A "NIC-based protocol address" is a type of network-interface-based protocol address that includes a plurality of identifiers that identify a sequence of network interface components. A "hop-based protocol address" is a type path-based protocol address since a hop is a type of network path.

[0066] Given the above definitions, note that the terms "network path" and "hop" may be defined in terms of network interfaces. A "network path" and a "hop path" include a sequence of network interfaces in a network that are included in transmitting data between a pair of path end nodes in the network. A "hop" refers to at least part of a network path that includes a pair of consecutive network interfaces in a sequence of network interfaces in a network path. A "network path" is thus a sequence of hops in a network that respectively include a sequence of pairs of consecutive network interfaces included in transmitting data from a first path end node of the network path to a second path end node of the network path.

[0067] The term "network topology" as used herein refers to a representation of protocol endpoints and/or nodes in a network, and representations of hops representing communicative couplings between and/or among the protocol endpoints and/or nodes in the network. A network may have different network topologies with respect to different network protocols. A network topology may represent physical communicative couplings between nodes in the network. A network topology may represent logical couplings between protocol endpoints and/or nodes of a particular network protocol or a particular type of network protocol.

[0068] The domain name system (DNS) of the Internet operates based on an application layer protocol defined by the DNS. The nodes in the DNS are communicatively coupled via the DNS protocol and may be represented by a logical network topology. A DNS system includes nodes connected via the DNS protocol and thus has a network topology defined by nodes that include protocol endpoints of the DNS protocol. In still another example, a token-ring network has a circular topology at the link layer, but may have a star topology at the physical layer.

[0069] As used herein, an "entity-specific address space" refers to an address space defined for a specific entity where the addresses in the address space operate as identifiers in the context of the entity. An address from an entity-specific address space is referred to herein as an "entity-specific address". An address is "entity-specific" in that what it identifies is based on the entity to which it is specific. Another address having the same form and content may identify a different entity when in an address space specific to another entity. Addresses in an entity-specific address space operate as identifiers in the context of an entity to which they are "specific" as defined by the specific association of the address space and the entity. Without knowledge of the entity to which an entity-specific address space is specific, what an address in the entity-specific address space identifies is indeterminate. The terms "entity-specific address" and "entity-specific identifier" are used interchangeably herein. An entity-specific address may identify an entity included in the entity to which the address is specific or may identify an entity external to the entity to which the address is specific. The fact that an address is entity-specific does not define a scope for the address.

[0070] A portion of a network is a type of entity. A type of entity-specific address space described herein is a scope-specific address space. As used herein, a "scope-specific address space", specific to a particular region of a network, is an address space defined for the particular network region, where an address in the scope-specific protocol address operates as identifier, according to a network protocol, of a protocol endpoint in a node outside of the particular region when processed in the context of a node in the particular region. The region is indicated by the span of an indicated scope. The terms "region" and "zone" are used interchangeably herein. An address from a scope-specific address space is referred to herein as a "scope-specific protocol address". An address is "scope-specific" in that what protocol endpoint it identifies depends on the region to which it is specific. Another address having the exact same form and content may identify a different protocol endpoint when in an address space that is specific to another region. A protocol address in a scope-specific address space serves as an identifier in the context of a node in a region to which the scope-specific address space is "specific" as defined by an association of the address space and the region indicated by the scope. Without knowledge of the particular region to which a scope-specific

address space is specific, what a scope-specific protocol address in the scope-specific address space identifies is indeterminate. The terms “scope-specific protocol address” and “scope-specific protocol identifier” are used interchangeably herein. Types of scope-specific address spaces indicating exemplary spans include site-specific, LAN-specific, subnet-specific, city-specific, business-specific, and node-specific.

[0071] For a network protocol, an address in a scope-specific address space serves as an identifier of a protocol endpoint in a node. Data may be received via the protocol endpoint from a network via one or more network interfaces that operatively couple the node to the network. Data may be sent via the protocol endpoint for transmitting over the network via the one or more network interfaces in the node. Since a protocol endpoint of a network protocol is included in a node and is accessible via a network via a network interface, a protocol address identifying the protocol endpoint also identifies the node and identifies a network interface of the node.

[0072] As used herein, a “node-specific address space” is a scope-specific address space defined for a specific node in a network, where the addresses in the node-specific address space operate as identifiers of nodes and/or network interfaces in the network when processed in the context of the specific node. An address from a node-specific address space is referred to herein as a “node-specific address”. An address is “node-specific” in that what it identifies depends on the node to which is defined as specific. Another address having the exact same form and content may identify a different node when in an address space specific to another node. Addresses in a node-specific address space operate as identifiers in the context of a node to which they are “specific” as defined by the specific association of the address space and the node. Without knowledge of the node to which a node-specific address space is specific, addresses in the node-specific address space are indeterminate. The terms “node-specific address” and “node-specific identifier” are used interchangeably herein. A node-specific address space is a type of scope-specific address space.

[0073] The term “node” is defined above. Note that an identifier of a network interface in a network also identifies a node that includes the network interface. Thus, a network interface-specific address is also a node-specific address. Network interfaces in a node may have their own respective network interface-specific address spaces that are also

node-specific. The network interface-specific address spaces may be combined to form a node-specific address space and/or may be managed as separate address spaces. The adjectives “node-specific” and “network interface-specific” may be used interchangeably.

[0074] A scope-specific identifier differs from a scoped address as described in "Request for Comments" (RFC) document RFC 4007 by S. Deering, et al, titled "IPv6 Scoped Address Architecture”, published by the IETF in December, 2006 and further described in Application No. 11/962,285, by the present inventor, filed on 2007/12/21, entitled “Methods and Systems for Sending Information to a Zone Included in an Internet Network”. A scoped address space is shared by nodes in a given scope. While a link-local scoped address is specific to a particular node, a link-local scoped address simply identifies a network interface component local to the particular node. A loop-back internet address is specific to a node as well. Neither link-local scoped addresses nor loop-back addresses identify one node to another. As such, neither serves as a node-specific identifier as defined above.

[0075] A “scoped address” is described by RFC 3513 and RFC 4007 as an identifier that, in a particular region of a network, serves as a protocol address of a network interface and/or a node in the particular region. The extent of the particular region is referred to as the scope of the region and thus the scope within which the identifier serves as a protocol address. A particular region included within a scope as indicated by its span. A scoped address is a valid protocol address only within a particular region as indicated by the address’s indicated scope. Examples of scope indicators include node-scope where identifiers are valid only to a single node in the indicated span, LAN-scope where identifiers are valid for nodes in the span of a particular LAN, and subnet-scope where identifiers are valid only for nodes in a particular subnet. RFC 3513 currently defines support for link-local scope, site-local scope, a global scope. A data unit transmitted with a scoped address should not be delivered to node that does not have a network interface in the span indicated by the scope.

[0076] “Path information” is any information that identifies a network path and/or hop path for data transmitted via one or more specified network protocols. Path information may be identified by identifying network interfaces, NICs, nodes, and/or hops included

in a network path. “Address information” is any information that identifies a protocol address that, for a network protocol, identifies a protocol endpoint. Address information may identify a unicast protocol address for a network protocol. In identifying a protocol endpoint, a protocol address identifies a node and a network interface. Those skilled in the art will understand upon reading the descriptions herein that the subject matter disclosed herein is not restricted to the network protocols described and/or their corresponding OSI layers. For ease of illustration, the subject matter is described in terms of protocols that correspond to OSI layer three, also referred to as network layer protocols, in general. Particular descriptions are based on versions of the Internet Protocol (IP). Address information may identify one or more protocol addresses. Exemplary protocol addresses include IP addresses, IPX addresses, DECNet addresses, VINES Internet Protocol addresses, and Datagram Delivery Protocol (DDP) addresses.

[0077] The term “path-based address” is defined above. A “node-based address” is a path-based address where some or all of the address includes node identifiers that identify a sequence of nodes in a network path. A “network-interface-based address” is a path-based address where some or all of the address identifies includes identifiers of a network interfaces in sequence in a network path. A “NIC-based address” is a type of network-interface-based address that identifies a sequence of network interface components. A “hop-based address” is a path-based address where some or all of the address identifies one or more hops in a network path. The protocol address types defined are not mutually exclusive.

[0078] The term “metric space”, as used herein, refers to a set, as defined in mathematics, where a distance between elements of the set is defined according to a metric. Metric spaces defined in Euclidean geometry are well-known examples. Those skilled in the art of metric spaces, such as Euclidian spaces, will appreciate that a one-to-one mapping may be determined and/or otherwise identified for mapping addresses from a first coordinate space having a first origin for a metric space to addresses from a second coordinate space having a second origin in the metric space. Given a mapping rule between a first scope-specific address space and a second scope-specific address space and a mapping between the second scope-specific address space and a third

scope-specific address space based on a third coordinate space identifying a third origin in the metric space, a mapping from the first coordinate space to the third coordinate space may be determined. A mapping between coordinate spaces for a metric space may be included a coordinate shift and/or a rotation, for example. The mapping may be pre-specified and accessible to the nodes in one or both address spaces. Mapping between locations in a number of different metric spaces is well-known in mathematics. For example, a top half of the surface of sphere may be mapped to a plane. Some will further appreciate that some metric spaces may be mapped to other metric spaces. Some of these mappings are one-to-one and onto.

[0079] An "interaction", as the term is used herein, refers to any activity including a user and an object where the object is a source of sensory data detected by the user. In an interaction the user directs attention to the object. An interaction may also include the object as a target of input from the user. The input from the user may be provided intentionally or unintentionally by the user. For example, a rock being held in the hand of a user is a target of input, both tactile and energy input, from the user. A portable electronic device is a type of object. In another example, a user looking at a portable electronic device is receiving sensory data from the portable electronic device whether the device is presenting an output via an output device or not. The user manipulating an input component of the portable electronic device exemplifies the device, as an input target, receiving input from the user. Note that the user in providing input is detecting sensory information from the portable electronic device provided that the user directs sufficient attention to be aware of the sensory information and provided that no disabilities prevent the user from processing the sensory information. An interaction may include an input from the user that is detected and/or otherwise sensed by the device. An interaction may include sensory information that is detected by a user included in the interaction that is presented by an output device included in the interaction.

[0080] As used herein "interaction information" refers to any information that identifies an interaction and/or otherwise provides data about an interaction between a user and an object, such as a portable electronic device. Exemplary interaction information may identify a user input for the object, a user-detectable output presented by an output

device of the object, a user-detectable attribute of the object, an operation performed by the object in response to a user, an operation performed by the object to present and/or otherwise produce a user-detectable output, and/or a measure of interaction. The term "operational component" of a device, as used herein, refers to a component included in performing an operation by the device.

[0081] Interaction information for one object may include and/or otherwise identify interaction information for another object. For example, a motion detector may detect user's head turn in the direction of a display of a portable electronic device. Interaction information identifying the user's head is facing the display may be received and/or used as interaction information for the portable electronic device indicating the user is receiving visual input from the display. The interaction information may serve to indicate a lack of user interaction with one or more other objects in directions from the user different than the detected direction, such as a person approaching the user from behind the user. Thus the interaction information may serve as interaction information for one or more different objects.

[0082] The term "attention information" as used herein refers to information that identifies an attention output and/or that includes an indication to present an attention output. Attention information may identify and/or may include presentation information that includes a representation of an attention output, in one aspect. In another aspect, attention output may include a request and/or one or more instructions for processing by a processor to present an attention output. The aspects described serve merely as examples based on the definition of attention information, and do not provide an exhaustive list of suitable forms and content of attention information.

[0083] As used herein the term "attention criterion" refers to a criterion that when met is defined as indicating that interaction between a user and an object is or maybe inadequate at a particular time and/or during a particular time period. In other words, the user is not directing adequate attention to the object.

[0084] As used herein, the term "tag" refers to a character string, which may include one or more words, which may be associated with a resource to create an association between the resource and another resource also tagged with the tag. Tags are often used in sharing media, social bookmarking, social news and blog entries to help users

search for associated content. In some contexts, the term “tagging” as used herein refers to the process of associating a tag with a resource that can be tagged. As used herein, the term “tagged association” refers to an association that identifies a tag and a resource that is tagged with the tag. A resource is said to be “tagged” with a tag when a tagging process, also referred to as “tagging”, has successfully created a tagged association. In other contexts, a “tagging” refers to a tagged association.

[0085] The term “matched tag”, as used herein, refers to an association that identifies a tag and a tagging criterion, where the tagging criterion is based on data other than the tag. The tagging criterion may be based additionally on the tag. The term “tagging association”, as used herein, refers to a matched tag represented in a memory, either persistent and/or volatile. For example, a tagging association may be realized as a record that includes and/or identifies a tag and a tagging criterion. The terms “matched tag” and “tagging association” are used interchangeably. When the term “matched tag” refers to something other than its tagging association, it will be made clear. For example, a reference to matched tag presented in a user interface clearly identifies a UI element that identifies and/or otherwise represents and/or otherwise identifies a matched tag either realized or to be realized in a tagging association. A tagged association, defined above, may identify a matched tag and a resource that is tagged with the tag identified by the matched tag. Such an association is also referred to herein as a “matched tag association”. A resource may be tagged with a matched tag when a tagging criterion identified by the matched tag is met for the resource.

[0086] As user herein, the term “vocabulary” refers to a set of valid values that can be assigned to and/or included in a data element. With respect to a tag, a vocabulary defines valid tags. A vocabulary may be specified by one or more rules and/or by identifying one or more valid values directly and/or indirectly.

[0087] As used herein, the term” dictionary” refers a vocabulary wherein one or more terms in the vocabulary is assigned a definition. A “definition” as the term is used herein refers to information that identifies semantic information about a data value, such as a t tag. Semantic information may include one or more of a textual description of a meaning of the term, an audio description, a visual description, and information identifying a use or context for the term. A use and/or context may be identified, for example, by

identifying a part of speech, a dialect or language, an antonym, a synonym, and/or an example usage of the defined data value.

[0088] As used herein, any reference to an entity "in" an association is equivalent to describing the object as "identified" by the association, unless explicitly indicated otherwise.

[0089] As used herein, the term "communication" refers to information including a message sent and/or for sending via a network between communicants. A message may include text data, audio data, and/or image data. The term "communicant" as used herein refers to a user included in a communication as a sender and/or a receiver of the information. A communicant is represented by a "communications agent" configured to operate in an execution environment to send data to and/or receive data from another communications agent, on behalf of the represented communicant, according to a communications protocol via network. A communications protocol defines and/or otherwise identifies an address space including communications addresses for delivering data sent in a communication from one communications agent to another. A communications protocol is a type of network protocol.

[0090] The term "communicant alias" as used herein refers to an identifier of a communicant in a communication where the communicant alias is not a communications address included in an address space of a communications protocol for sending and/or receiving data in the communication.

[0091] The term "attachment" as used herein refers to a portion of a communication that includes data from one communicant to another other than data in the message portion. A resource sent as an attachment is data that is typically not presented "inline" or in a message included in a message portion of a communication. Email attachments are perhaps the most widely known attachments included in communications. An email attachment is a file or other resource sent along with an email in a portion of the email separate from a message portion. A communication may include one or more resources as one or more attachments.

[0092] The terms "contactor" and "contactee" identify roles in a communication. A "contactor" provides information for identifying a "contactee" in a communication. A contactee may be included in a communication by a contactor and/or the contactor's

communications agent. A contactor and/or a contactee are roles of a user or communicant of a communications agent and may played by a living being, a node, a component, and/or an application. Both a contactor and a contactee are communicants in a communication.

[0093] As used here, the term "tagging- attribute association" refers to an association that identifies one or more tagging criterion and a resource attribute that the one or more tagging criterion are each based on. That is, whether a tagging criterion identified in a tagging-attribute association is met for a specified resource depends on an attribute of the resource identified by the tagging-attribute association. The term attribute, as used herein, includes the resource, as a resource is considered to be an attribute of itself and/or otherwise to include at least some of its attributes.

[0094] As used herein, the term "resource scope" refers a bounded space in which a resource may be identified. The space may be any space in which a resource may be locate and/or otherwise identified. A bounded may be may include one or more of an identifier space such as file system identifier space and/or a URI address space, a storage space, a geospatial space, a demographic space, and a data type – to name a few examples. space, defines a set of resources from which resources may be identified. For example, the files within a specified folder and its subfolders in a file system define and/or otherwise specify a resource scope. All user accounts that have an administrator security role define another resource scope. The present disclosure describes a resource scope with respect to resources tagged with a specified tag that makeup a set, referred to herein as a "tagged resource set". A tagged resource set may include a resource tagged with a tag identified by a matched tag.

[0095] In specifying a resource scope, one or more criterion may be specified. The terms "criterion" and "constraint" are used interchangeably herein, with respect to a resource scope. A criterion included in specifying a resource scope is referred to herein as a "scope criterion". A scope criterion may be expressed via any suitable manner, such as a file name, a wildcard expression, a regular expression, a data store location, a data base query, geospatial coordinates, a time, a date, a duration, a relationship to a user, and/or an address – too name a few examples. With respect to a tag, a scope criterion included in defining a resource scope differs from a tagging criterion for the tag

in that the scope criterion included in defining a resource scope need not be met in order for a resource to be tagged with the tag. Further a resource scope may be used for purposes other than processing related to a tagged resource.

[0096] One or more scope criterion that together specify and/or otherwise define a resource scope is/are referred to herein as a "scope descriptor". When all the criteria in a scope descriptor are met for a resource, the resource is in the resource scope defined by the scope descriptor. The one or more separate criterion in a scope descriptor may be evaluated in any order suitable or desirable in any given embodiment. In an aspect described below, one scope criterion, that may be included in a scope descriptor, may be specified to determine and/or otherwise identify when the scope descriptor is "active". A scope criterion that indicates whether a scope descriptor is active is referred to herein as a "scope condition". In a further aspect a scope descriptor may include no other criterion indicating the resource scope includes all resources accessible to an application, subsystem, execution environment, and/or group of any of the preceding that are processing and/or otherwise making use of the scope descriptor. Those skilled in the art will understand that whether a scope criterion in a scope descriptor is called a scope condition or not depends on the role played by the scope criterion. The subject matter may be described without referring to a scope criterion in the role of a scope condition without limiting the scope of subject matter.

[0097] Although, those skilled in the art will know that a scope criterion may be defined as "met" when it is true or may be defined as "met" when it is not true; for purposes of the present disclosure, A scope criterion is said to be "met" for a resource when it indicates the resource is in the resource scope defined by the scope descriptor, and is to be "not met" otherwise. A scope criterion in the role of a scope condition in a scope descriptor is said to be "met" when it indicates the scope descriptor is active, as is said to be "not met" otherwise. In an aspect, a scope descriptor may not include a scope condition or may include a scope condition that is always met. Such a scope descriptor is active by definition. Whether a criterion, in a scope descriptor is met or is not met, may depend on an evaluation of the scope criterion that is based on an identifiable entity. For a scope condition, such an entity is referred to herein as a "condition entity". More generally, for a scope criterion, such an entity is referred to herein as a "scoping

entity". Exemplary entities include a user, a document, a measure of time, a location, and the like.

[0098] An association between a tag and scope descriptor is referred to herein as a "scoped tag". A scoped tag is also referred to herein as a match set. As defined above, a tagging criterion may be used to determine whether to tag a particular resource with a particular tag. The tagging criterion is associated with the tag by a matched tag. A scope descriptor identifies resource scope that includes a set of resources. Scope descriptors are necessarily associated with resources.

[0099] A "tagging context", as the term is used herein, identifies and/or otherwise includes a set of tagged associations or taggings. The taggings may identify one or more tags and one or more resources. A "tagging context specification", as the term is used herein, is data that defines and/or otherwise specifies a tagging context. A tagging context specification may be stored in a memory location. The memory location may be contiguous or not and may be included in one or more data storage media, which may be volatile or not. A tagging context specification may include and/or otherwise identify any suitable criterion, referred to in general as "context criterion", for determining whether a tagging is in a tagging context defined by the tagging context specification. For example, a tag contact specification may identify one or more scope descriptors, one or more tags some or all of which may be and/or may be identified by matched tags, and a context condition; all of which are context criterion by definition.

[0100] A "context condition", as the term is used herein, refers to a context criterion that when met indicates that a tagging context is active for one or more taggings. When a context condition is not met, it is said to be inactive or not active for one or more taggings. For example, a context condition in a tagging context specification may be met for a first user. Taggings in the tagging context are "active" for the first user. For a second user, the context condition may not be met. The taggings are not active for the second user. A context condition, or more generally a context criterion, may be based on any suitable data for any suitable entity. A tagging context may be active for a task and not for another and/or a tagging context may be active for one location and not for another, to name a couple of other examples. When a context condition indicates that

an associated tagging context is active for a tag, the context condition is herein said to be "met", otherwise the context condition is not met.

[0101] FIG. **3A** illustrates an arrangement of components in a system that operates in an execution environment, such as execution environment **102** in FIG. **1**. The arrangement of components in the system operates to perform the method illustrated in FIG. **2A**. The system illustrated includes a query handler component **302**, a identification request constructor (IRC) component **304**, and a result handler component **306**. A suitable execution environment includes a processor, such as processor **104**, to process an instruction in at least one of a query handler component, a identification request constructor component, and a result handler component. FIG. **3B** illustrates an arrangement of components in a system that operates to perform the method illustrated in FIG. **2B**. The system illustrated includes a content manager component **309**, a query handler component **302**, a query director component **312**, and a identification response builder (IRB) component **314**. A suitable execution environment includes a processor, such as processor **104**, to process an instruction in at least one of a content manager component, a query handler component, a query director component, and a identification response builder (IRB) component.

[0100] Some components, illustrated in the drawings are identified by numbers with an alphanumeric suffix. A component may be referred to generically in the singular or the plural by dropping a suffix of a portion thereof of the component's identifier. For example, window UI elements, such as a window UI element **602a** in FIG. **6A**, a window UI element **602b** in FIG. **6B**, and their adaptations and analogs; are referred to herein generically as a window UI element **602** or window UI elements **602** when describing more than one. Other components identified with an alphanumeric suffix may be referred to generically or as a group in a similar manner.

[0101] The arrangement components illustrated in FIG. **4** may be adapted to operate in a number of execution environments to perform the method illustrated in FIG. **2A** and/or the method illustrated in FIG. **2B**. FIG. **4** is a block diagram illustrating the components of FIG. **3A**, FIG. **3B**, and/or analogs of the components of FIG. **3A** and FIG. **3B**, respectively adapted to operate in an execution environment **401** that includes and/or otherwise is provided by one or more nodes. FIG. **1** illustrates key components of an

exemplary device that may at least partially provide and/or otherwise be included in an execution environment. The components illustrated in FIG. **4** may be included in or otherwise combined with the components of FIG. **1** to create a variety of arrangements of components according to the subject matter described herein.

[0102] As stated, the various adaptations of the arrangement in FIG. **3A** as well as the various adaptations of the arrangement in FIG. **3B** illustrated and described herein are not exhaustive. For example, those skilled in the art will see based on the description herein that arrangements of components for performing the method illustrated in FIG. **2A** and the method illustrated in FIG. **2B** may each be distributed across more than one node and/or execution environment. For example, such an arrangements may operate at least partially in a browser in a one node and at least partially in a server in another node interoperating via a network.

[0103] FIG. **5** illustrates a first node **502** and an execution environment **401** of second node **504** as exemplary devices that each may be included in and/or otherwise adapted for providing an instance, adaptation, and/or analog of execution environment **401** in FIG. **4**. As illustrated in FIG. **5**, first node **502** and an execution environment **401** of second node **504** are operatively coupled to a network **506** via respective network interface components enabling first node **502** and execution environment **401** of second node **504** to exchange data in a communication on behalf of communicants represented by the respective nodes.

[0104] FIG. **4** illustrates communications agent **403** including an adaptation of the arrangement of components in FIG. **3A** and an adaptation of the arrangement of components in FIG. **3B**. A first communications agent **403** may operate, in first node **502**, on behalf of a first communicant to communicate with a second communicant represented by a second communications agent **403** operating in execution environment **401** of second node **504**.

[0105] Execution environment **401** is illustrated in FIG. **4** hosting communications agent **403**. Components in FIG. **4** are referenced for illustrative purposes in describing communications agents operating in an execution environment of first node **502** and an execution environment of execution environment **401** of second node **504**. Exemplary communications agents include email clients, phone clients including Voice over

Internet Protocol (VoIP) clients, instant messaging clients, short message service (SMS) clients, multimedia message service (MMS clients), multi-media communications clients including video phone clients, and other data transfer agents.

[0106] Communications agents **403** in FIG. **4**, respectively operating in first node **502** and in execution environment **401** of second node **504** in FIG. **5** may interoperate via respective network stacks **405**. Communications agents **403** may communicate via one or more communications protocols. FIG. **4** illustrates communications protocol component **407** exemplifying a subsystem for exchanging data via network **506** according to one or more communications protocols, such as simple mail transfer protocol (SMTP), an instant messaging protocol, and/or a real-time voice and/or video protocol. A communication between communications agents **403** in first node **502** and execution environment **401** of second node **504** may include more than one type of data and may use one or more communications protocols in exchanging one or more types of data via network **506**.

[0107] Instances, adaptations, and/or analogs of communications agent **403** in FIG. **4**, in first node **502** and in execution environment **401** of second node **504**, respectively, may communicate via discrete messages, a request/reply protocol, a data streaming protocol, a session and/or connection-oriented protocol, a connectionless protocol, a real-time communications protocol, an asynchronous communication, a store and forward communications protocol, a reliable delivery communications protocol, a best-effort delivery communications protocol, and/or a secure protocol, to name a few communications options.

[0108] FIG. **4** illustrates communications agent **403** including a content manager component **409**. Content manager component **409** may interoperate with communications protocol layer component **407** and/or network stack **405** to receive data in one or more communications via network **506** with another communications agent in another node. Content manager component **409** is operatively coupled, via com-in component **411**, to communications protocol component **407** for receiving the data from the other node.

[0109] Data received in a communication may include one or more resources and/or content types. Exemplary content types include plain text, markup such as hypertext

markup language (HTML), audio data, image data, and/or executable data. Executable data may include script instruction(s), byte code, and/or machine code. In FIG. **4**, communications agent **403** includes one or more content handler components **413** to process data received according to its content type. A data type may be identified by a MIME type identifier. Exemplary content handler components **413** include a text/html content handler component for processing HTML representations; an application/xmpp-xml content handler component for processing extensible messaging and presence protocol (XMPP) streams including presence tuples, instant messages, and audio content handlers including and/or configured to retrieve suitable codices; one or more video content handler components for processing video representations of various types; and still image data content handler components for processing various image data representations.

[0110] Content handler component(s) **413** process received data representations and may provide transformed data from the representations to one or more user interface element handler components **415**. One or more user interface element handler components **415** are illustrated in a presentation controller component **417** in FIG. **4**. Presentation controller component **417** may manage visual, audio, and other types of output for its including application as well as receive and route detected user and other inputs to components and extensions of its including application, communications agent **409**. A user interface element handler component **415** may be adapted to operate at least partially in a content handler component **413** such as a text/html content handler component and/or a script content handler component. Additionally or alternatively, a user interface element handler component in an execution environment **401** may be received in a communication. For example, a communication, such as an email, may include an HTML content type portion and a script content type portion.

[0111] FIG. **6A-G** illustrates various communications agent windows **602** presentable in a presentation space of a display device, such as output device **130** in FIG. **1**. In FIG. **6a**, a communications agent window **602a** includes a contactor user interface (UI) element **604a** for presenting an identifier of a communicant in the role of a contactor in a communication represented by the communications agent window **602a**. A communications agent window **602a** also includes a contactee UI element **606** for

presenting one or more contactee identifier(s) identifying one or more communicants in the role of contactee(s) included in the communication. A presentation space **608** is provided by each communications agent window **602a** for presenting a user message UI element **610** for presenting a message addressed to one or more contactees identified in a contactee UI element **606a**. A presentation space **608a** may also be provided for presenting one or more UI controls for exchanging data in and/or otherwise managing a communication. A send UI element **612a** illustrates an exemplary UI element that may correspond to user input to send data in a communication to one or more identified contactees.

[0112] Data to send in a communication to a communications agent in execution environment **401** of second node **504** may be received by one or more content handler component(s) **413** operating in first node **502** to transform the data into one or more data representations suitable for transmitting in the communication and/or suitable for processing by the communications agent in execution environment **401** of second node **504**. The one or more data representations may be provided to content manager component **409** for sending in the communication to execution environment **401** of second node **504**. Content manager component **409** may package the one or more data representations in a message formatted according to a communications protocol of the communications agent. Communications protocol component **407** may send the data according to the specification(s) of the communications protocol. Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in the communication to a communications agent via a network.

[0113] Content manager component **409** operating in execution environment **401** included in and/or otherwise provided by first node **502** may provide the packaged, encoded, and/or transformed data to communications protocol component **407** via a com-out component **419**. Com-out component **419** as described above operatively couples communications agent **409** to communications protocol component **407** according to an interface provided by communications protocol component **407** for sending data in a communication according to a communications protocol.

Communications protocol component **407** may further package and/or otherwise transform the data to send via network stack **405** for delivery via network **506** to execution environment **401** of second node **504**.

[0114] As described above, a communications protocol may operate via one or more nodes in a network in a network path including a first node and an execution environment **401** of second node **504**. Exemplary path nodes include mail relay nodes, phone switch nodes, and proxy nodes such as instant messaging proxies for communicating through firewalls. As indicated, path node **508** illustrates such a node.

[0115] Sending data in a communication requires identifying a contactee. A contactee may be represented by a communications address in an address space of a communications protocol. In one aspect, information identifying a communications address may be received from a user/communicant of a communications agent in a user node. In FIG. **4**, presentation controller **417** and/or a UI element handler **415**, presenting and/or managing interaction with contactor UI element **604** in FIG. **6A** for first node **502**, may receive a contactor alias in response to a user input corresponding to UI element handler **604**. The user/communicant of first node **502** may enter a contactor alias, such as “Dad”, via a keyboard and/or select a predefined communicant alias presented in a selection UI control element via a UI element handler component **415**. The user input may be detected by input driver **421**. Corresponding input information may be routed to presentation controller **417** by GUI subsystem **423**. GUI subsystem **423** may send presentation information to a display device via a graphics subsystem **425**. Communications agent **403** may identify a communication address associated with the contactor alias, “Dad”. Other communications address(es), such as for one or more contactees, may be received similarly and/or in any suitable manner.

[0116] Data may be sent in a communication according to a form or type of the communication and/or other attribute of the communication such as a security attribute, the amount of data to be set, a priority setting, a task setting, and the like. Some forms of communication do not require a session and/or connection between a first node and an execution environment **401** of second node **504** in order to send data between the two nodes, while others do. An email and/or instant message may use a store and forward model of delivery.

[0117] Data may be sent in a communication in response to a communicant input. A contactor may provide an input corresponding to send UI element **612a** in FIG. **6A**. The input may be received by presentation controller component **417** and/or one or more UI element handlers **415** corresponding to send UI element **612a**. In response to detecting the input, presentation controller component **417** may provide data to be sent in the communication to one or more content handler components **413** according to the content type(s) of the data to be sent.

[0118] The one or more content handler components **413** may encode, format, and/or otherwise transform the data for sending in a message, such as an email message. The one or more content handler components **413** may provide data to be sent to content manager **409**, instructing content manager component **409** to send the data in the communication for delivery to a communications agent in execution environment **401** of second node **504**. Content manager component **409** interoperating with com-out component **419** may further format and/or transform the data for sending in the communication according to communications protocol, for example according to an email communications protocol, by communications protocol component **407**. Communications protocol component **407** may send the communication for delivery to the communications agent in execution environment **401** of second node **504** via network **506**.

[0119] Path node **508** may relay data sent in the communication between first node **502** and execution environment **401** of second node **504**. Path node **508** may determine a next node and/or a network interface in a network path communicatively coupling first node **502** and execution environment **401** of second node **504** for exchanging data in a communication between communications agents **403** in the first node **502** and in the execution environment **401** of second node **504**.

[0120] For session-oriented and/or connection-oriented communication a session and/or connection may be established if a session/connection has not already been established. Data may be sent for delivery to a communications agent identified based on a contactee communications address during session and/or connection setup. For example, for a voice communication a voice communication may be established via a session initiation protocol. Communications protocol component **407** may operate

according to the session initiation protocol specifications. Communications protocol component **407** operating in first node **502** may locate a communications agent by communicating with one or more nodes in network **506** according to the session initiation protocol. Communications protocol component **407** may locate a communications agent in execution environment **401** of second node **504**, based on a communications address for the contactee located based on the contactor alias.

[0121] Once a communication session is established, such as a voice session, data may be sent according to the session communications protocol, such as RTP. Data may be sent according to the session initiation protocol in the communication for managing the voice communication session and/or for exchanging text, image, and/or other data outside of the voice session. Path node **508** may be included in session and/or connection setup. Alternatively or additionally, path node **508** may be included in a network path in a session and/or connection.

[0122] With reference to FIG. **2A**, a block **202** illustrates that the method includes receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. Accordingly, a system for processing a data object identification request in a communication includes means for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. For example, the arrangement in FIG. **3A**, includes query handler component **302** that is operable for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion. FIG. **4** illustrates query handler component **402** as an adaptation and/or analog of the query handler component **302** in FIG. **3A**. One or more query handler components **402** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, by a first communications agent in a first execution environment from a first user represented by the first communications agent, data object

information identifying a data object matching criterion. In FIG. **4**, query handler component **402** is illustrated as a component of communications agent **403**.

[0123] FIG. **7** includes dataflow message **702** that illustrates a data flow internal to execution environment **401** of first node **502** and/or a message received via network **506** by execution environment **401** of first node **502**. Dataflow message **702** illustrates a data object information dataflow to query handler component **402** operating in execution environment **401** of first node **502**. Data object information is identified to and/or otherwise received by query handler component **402**.

[0124] In various aspects, a query handler component **402** may receive data object information from a user via one or more UI element handler components **415**. The data object information may be received, in response to a user input detected by an input device of an execution environment **401** that includes the query handler component **402**. Alternatively or additionally, data object information may be received by a query handler component in a message received via a network. For example, a user may identify data object information to a browser operating in a client node. The browser may send the data object information via a network communicatively coupling the client node to a web service provider node. The web service provider node may include a query handler component that may operate to receive the data object information identified in the message from the browser. FIG. **6A** illustrates an exemplary user interface that may be presented by communications agent **403** operating in execution environment **401** of first node **502**. A UI element handler component **415** may operate to present a user interface element to allow a user to identify data object information. A data object information (DOI) UI element **616a** is illustrated in FIG. **6A** as an example. The same or different UI element handler component **415** may present various other UI elements included in DOI UI element **616a**. DOI UI element **616a** and one or more UI elements it includes may be presented by one or more corresponding UI element handler component(s) **415** that may operate based on a schema that defines valid data object information and/or data object identification requests for including in a communication via a communications agent **403**. The schema defines one or more rules and/or a vocabulary that defines whether data object information and/or a data object identification request is valid. Those skilled in the art will understand that

numerous languages and/or schemas for providing data object information and/or data object identification requests currently exist including relatively simple key word based queries to relatively more complicated regular expression languages and data base query languages, such as various versions of structured query language (SQL). New schemas are and will be created, and are, thus, considered within the scope of the subject matter described herein.

[0125] DOI UI element **616a** illustrates a user interface for a keyword based schema. Match textbox UI element **618a** illustrates a textbox allowing a user to enter a keyword expression. Various checkbox UI elements **620a** allow a user to define a scope of a data object identification request by identifying whether a resulting data object identification request is to be applied to folders, files, and/or content of files in a file system and/or other data store of a receiving execution environment. A communications agent **403** may support more than one data object identification request schema and/or corresponding data object information schema. Thus. a communications agent may provide a user interface to receive valid data object information for each respective schema.

[0126] Returning to FIG. **2A**, a block **204** illustrates that the method further includes sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. Accordingly, a system for processing a data object identification request in a communication includes means for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. For example, the arrangement in FIG. **3A**, includes identification request constructor component 304 that is operable for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the

data object matching criterion, wherein the first message is addressed to the second user. FIG. **4** illustrates identification request constructor component **404** as an adaptation and/or analog of identification request constructor component **304** in FIG. **3A**. One or more identification request constructor components **404** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for sending, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user. In FIG. **4**, identification request constructor component **404** is illustrated as a component of communications agent **403**.

[0127] FIG. **7** includes dataflow message **704** that illustrates a data flow internal to execution environment **401** of first node **502** and/or message received via network **506** by execution environment **401** of first node **502**. Dataflow message **704** illustrates a create data object identification request (CIReq) dataflow to an IRC component **404** operating in execution environment **401** of first node **502**. Data object information may be identified to and/or otherwise provided to IRC component **404**. A query handler component **402**, operating as described with respect to block **202** of FIG. **2A**, may identify and/or otherwise provide data object information to an IRC component **404**.

[0128] A query handler component may be a type of content handler component that operates to process data object information to create a data object identification request that conforms to a schema for creating and/or otherwise constructing a valid data object identification request. The IRC component **404** may construct and/or otherwise create a data object identification request based on a schema that defines and/or otherwise identifies a valid data object identification request and/or a valid data object identification response for a particular type of communication supported by a communications agent. The data object identification response is constructed and/or otherwise created, based on the data object information, by the IRC component **404**. IRC component **404** in execution environment **401** of first node **502** may provide the

data object identification request to content manager component **409** to include and/or otherwise identify the data object identification request in a communication with execution environment **401** of second node **504**, such as email and/or instant message. Content manager component **409** may interoperate with other types of content handler components **413** to create and/or otherwise construct a message that includes the valid data object identification request.

[0129] In an aspect, the IRC component **404** may interoperate with the content manager component **409** to create the valid data object identification request according to a specified schema for the message, so that the message is valid according to a communications protocol supported by a communications protocol component **407** that operates to send the message. As described above, the content manager component **409** may operate in an execution environment **401** of first node **502** along with an IRC component **404** to transform data object information into a data object identification request to include along with data for other parts of a communication into one or more representations suitable for transmitting in a communication to another node, such as execution environment **401** of second node **504**. Some or all of the representations transmitted are suitable for processing by the communications agent in execution environment **401** of second node **504**. The content manager component **409** in the execution environment **401** of first node **502** may package the one or more data representations including a representation of the data object identification request into a message formatted according to the communications protocol.

[0130] In FIG. **8A**, a portion of an email communication **800a** is illustrated formatted as a multipart/mixed content type including data object identification request portion **802a**. A data object identification request portion of a communication may be identified as a data object identification request by its location in the communication and/or by an identifier or markup element, such as a MIME type identifier. A data object identification request may be detected based on content included in the message and/or based on metadata such as content-type header **804a** identifying a MIME type identifier, such as “application/id-request”, which may be defined for representing one or more matching criteria in a data object identification request. The “application/id-request” MIME type

identifier is exemplary. Other MIME type identifiers exist and/or may be defined to identify a data object identification request in a communication.

[0131] An IRC **404** and/or a content manager component **409** may operate to construct a data object identification request in a communication based on XQuery, regular expression, and/or SQL content – to name a few examples. A content type identifier may be included in a position and/or location that identifies a data object identification request in a communication. The position or location may be absolute or relative. For example, a schema for a communication may define that a data object identification request in a communication is included in the communication at the end of the communication. There may be one or more data object identification requests at the end. In another aspect, a schema for a communication may specify that a portion of a communication following a particular type of message portion is a data object identification request. Other data object identification requests may follow. If no data object identification request is included, the data object identification request portion may include no content or may include an indicator that no data object identification request is included.

[0132] FIG. **8A** illustrates an "application/id-request" MIME type identifier that may be defined to identify a schema for an XML-based language for specifying id-request XML documents. FIG. **8A** illustrates id-request document **806a**. Id-request document **806a**, as illustrated, includes criterion tag elements **808a** corresponding to the form elements in the user interface illustrated in FIG. **6A**. A criterion tag element **808a** identifies a data object matching criterion name, such as "query" indicating that the criterion is a query expression or a portion thereof. A "type" attribute identifies a schema with the identifier "keyword" for the expression. The criterion tag element **808a** identifies a value for a query "Uncle Joe" in a match-expression attribute. Another criterion tag element **808a** specifies a scope query specified in the first criterion tag element. FIG. **8A** illustrates an "and" tag **810a** indicating that all the matching criteria must be met in identifying a data object. An "or" tag (not shown) may be defined by a schema for id-request documents. Other operator elements and operator precedence may be defined by the schema. Grouping elements for managing operator precedence, such as a parenthesis element, may be defined by the schema.

[0133] FIG. **7** includes dataflow message **706** that illustrates a first message, including a data object identification request, sent via network **506** by execution environment **401** of first node **502** to execution environment **401** of second node **504**.

[0134] A data object identification request generated by an IRC component **404**, along with a user message and any other data to include in a communication, may be provided and/or otherwise identified to content manager component **409** for sending in the communication. The content manager component **409** in execution environment **401** of first node **502** may package the data including a representation of the data object identification request into the message formatted according to the communications protocol. Com-out component **419** may provide the data object identification request, the user message, and any other data for sending in the communication in representations suitable for sending by communications protocol component **407** to the communications agent in execution environment **401** of second node **504**. Communications protocol component **407** operating in execution environment **401** of first node **502** may send the data as and/or in a communication, such as an email message, according to the specification(s) of the communications protocol. Communications protocol component **407** may further package and/or otherwise transform the data to send via network stack **405** for delivery via network **506** to execution environment **401** of second node **504**.

[0135] Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in a communication with the communications agent in execution environment **401** of second node **504**.

[0136] Returning to FIG. **2A**, a block **206** illustrates that the method yet further includes receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. Accordingly, a system for processing a data object identification request in a communication includes means for

receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. For example, the arrangement in FIG. **3A**, includes result handler component **306** that is operable for receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. FIG. **4** illustrate result handler components **406** as an adaptations and/or analogs of result handler component **306** in FIG. **3A**. One or more result handler components **406** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, by the first communications agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message. In FIG. **4**, a result handler component **406** is illustrated as a component of application **403**.

[0137] FIG. **7** includes dataflow message **714** that illustrates a second message, including a data object identification response, received via network **506** by execution environment **401** of first node **502** from execution environment **401** of second node **504**.

[0138] In addition to packaging communication information including a data object identification request into a valid communication, content manager component **409** operating in execution environment **401** of first node **502** may interoperate with communications protocol layer component **407** and/or network stack **405** to receive data in one or more communications via network **506** with another communications agent in another node. A result handler component **406** may receive some or all of the

communication in receiving and/or otherwise identifying a data object identification response in the communication. Content manager component **409** in FIG. **4** operatively couples result handler component **406** with network **506**, via com-in component **411**, communications protocol component **407**, and/or network stack **405** to receive the data from the other node. Thus, execution environment **401** of first node **502** may receive a communication, via content manager component **409** and result handler component **406**, that includes a data object identification response that is a response to a data object identification request detected by execution environment **401** of second node **504** in a message received previously from the execution environment **401** of first node **502**.

[0139] Com-in component **411** may provide data received in a communication to content manager component **409**. Content handler component **409** may identify a data object identification response in the communication sent from the execution environment **401** of second node **504**. Content manager component **409** may interoperate with result handler component **406** in identifying the data object identification response. The data object identification response may be identified by content manager component **409** according to a schema for the communication. For example, a portion of an email message may include a MIME-type identifier that identifies a corresponding part of the email as a data object identification response and/or otherwise identifies the part for routing to a result handler component **406** in communications agent **403** in execution environment **401** of first node **502** for detecting the data object identification response.

[0140] Data sent in second message **714**, by execution environment **401** of second node **504**, may be received by com-in component **411** operating in an instance, adaptation, and/or analog of execution environment **401** including and/or provided by first node **502**. The received communication includes a message addressed to the a communicant represented by communications agent **403** in first node **502** and a data object identification response identifying a data object located and/or otherwise identified based on the data object identification request identified by first message **706**. Content manager component **409** may detect the user message, such as message portion **802b** in second message **714**. Content manager component **409** may detect a data object identification response, such as data object identification response portion

804b in second message **714**. Message portion **802b** and data object identification response portion **804b** may be provided to suitable content handler components **413** based on the content types of the message portion and the data object identification response portion detected by content manager component **409**. In FIG. **4**, data object identification response portion **804b** may be provided to result handler component **406** as a content handler that matches the MIME type field **806b** illustrated in FIG. **8A**.

[0141] In an aspect, a data object identification response that is a response to a data object identification request sent in a previous message may include information identifying one or more data objects located and/or otherwise identified by the execution environment **401** of second node **504** that processed the data object identification request. For example, a data object identification response may be sent that identifies a URI for each data object located and/or otherwise identified by a first node in response to processing a corresponding data object identification request.

[0142] Returning to FIG. **2A**, a block **208** illustrates that the method yet further includes sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. Accordingly, a system for processing a data object identification request in a communication includes means for sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. For example, the arrangement in FIG. **3A**, includes access request constructor component **313** that is operable for sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. FIG. **4** illustrates access request constructor component **408** as an adaptation and/or analog of access request constructor component **308** in FIG. **3A**. One or more access request constructor components **408** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for sending, based on the data object identification response, a third message, via the network by the first execution environment, to access the data object. In FIG. **4**, a access request

constructor component **408** is illustrated as a component of result handler component **406**.

[0143] FIG. **7** includes dataflow message **716** that illustrates a present message that may be internal to execution environment **401** of first node **502** and/or may be a message sent to a remote device, such as a node including a browser, to send presentation information to present, via an output device, a representation of the data object identification response received in second message **714**. FIG. **7** also includes dataflow message **718** that illustrates a create access request message that may be a data flow internal to execution environment **401** of first node **502** and/or a message received via network **506** by execution environment **401** of first node **502**. In the dataflow message **718** an ARC component **408** operating in execution environment **401** of first node **502** may be invoked to operate based on result information. Result information may be identified to and/or otherwise by ARC component **408** as part of and/or in response to being invoked.

[0144] Result handler component **406** may be a type of content handler component **413** that operates to process data representations and may provide transformed data from the representations to one or more user interface element handler components **415**. Result content handler component **406** may operate to process data object identification responses according to a schema defining valid data object identification responses. Content manager component **409** may provide and/or otherwise identify various portions of a communication, such as a data object identification response, to one or more content handler components **413**. The data object identification response may be provided to result handler component **404**. In an aspect, result handler component may interoperate with the presentation controller component **417** in the execution environment **401** of first node **502** to present the data received in the data object identification response and/or other data received in the communication.

[0145] FIG. **6B** illustrates a response window UI element **602b** as an exemplary user interface that may be presented by communications agent **403** operating in execution environment **401** of first node **502** to present data received in a communication from execution environment **401** of second node **504**. FIG. **6B** illustrates a contactor UI element **604b**, a contactee UI element **606b**, a presentation space **608b** of the

response window UI element **602b**. Presentation space **608** includes a user message UI element **610b**, analogous to similar UI elements described with respect to FIG. **6A**. The various UI elements may be presented by one or more UI element handler components **415** in the communications agent **402** in execution environment **401** of first node **502**. A reply UI element **622b** and a reply all UI element **624b** illustrate UI elements in presentation space **608b** presented to allow a user of the execution environment **401** of first node **502** to respond and/or otherwise instruct execution environment **401** of first node **502** to perform an operation based on the message received in the communication. FIG. **6B** illustrates a data object identification requested query UI interface element **626b** where the presented term "None" may be defined to indicate no data object identification request was detected by the content manager component **409** in execution environment **401** of first node **502** in the communication received by execution environment **401** of first node **502** from second node **504** via network **506**.

[0146] Response window UI element **602b** in FIG. **6B** includes a data object identification response UI element **628b** for receiving user input to display a representation of one or more data objects located and/or otherwise identified by the execution environment **401** of second node **504** and identified in the data object identification response. One or more of the data objects is not included in the message received in the communication. In response to an input directed to data object identification response UI element **628b**, a corresponding UI element handler **415** may interoperate with result handler component **406** to present a located data resources dialog UI element **630b** that includes a presentation space **632b** for presenting a user detectable representation of some or all of the data objects identified in the received data object identification response from the execution environment **401** of second node **504**. In an aspect, one or more data objects located and/or otherwise identified by execution environment **401** of second node **504** may be included in a communication as attachments. In a further aspect, located and/or otherwise identified data objects not included in the message may be identified as described with respect to located and/or otherwise identified data objects dialog UI element **630b** and/or in any other suitable manner. In yet another aspect, files and/or other data entities identified in the data

object identification response may be illustrated in a manner similar to attachments rather than in a separate UI element as in FIG. **6B**, and accessed from first node **504** via network **506** as described below.

[0147] FIG. **6B** illustrates an access UI element **634b** and an access all UI element **636b** allowing a user to provide user input(s) to instruct communications agent **403** in execution environment **401** of first node **502** to request one or more data objects identified in the data object identification response.

[0148] FIG. **7** includes a dataflow message **720** that illustrates a third message that includes an access request to access a data object identified in the data object identification response received in second message **714**. FIG. **7** illustrates the access request is sent via network **506** by execution environment **401** of first node **502** to execution environment **401** of second node **504**.

[0149] In response to receiving second message **714**, execution environment **401** of first node **502** may present, in a manner described above, data received in the message to a user of execution environment **401** of first node **502** as described with respect to FIG. **6C**. Context menu **646c** provides UI elements to allow the user to instruct execution environment **401** of first node **502** to retrieve a data object identified in a data object identification response received from execution environment **401** of second node **504**. In response to a user input corresponding to a context menu item included in context menu **646c** a corresponding UI element handler component **415** may invoke ARC component **408** to construct an access request to retrieve one or more data objects identified in the data object identification response. FIG. **6G** illustrates another exemplary user interface that execution environment **401** of first node **502** may present to the user to allow the user to instruct execution environment **401** of first node **502** to construct and send a request. Auto-response window **602g** may be presented in response to a data object identification response received in a communication without a user message. Rather than a UI element for displaying a user message, a results pane UI element **670g** may be presented that identifies data objects located and/or otherwise identified by execution environment **401** of second node **504** that may be retrieved by execution environment **401** of first node **502**. In an aspect, execution environment **401** of second node **504** may process a data object identification request and automatically

send a data object identification response without involving the user of execution environment **401** of second node **504**. The data object identification response may be received without a user message as FIG. **6G** illustrates or may be received with a pre-written and/or automatically generated message. UI elements **672g-674g** may be presented by execution environment **401** of first node **502** to allow the user of execution environment **401** of first node **502** to request the retrieval of one or more data objects identified in the data object identification response.

[0150] ARC component **408** may be invoked to construct an access request to retrieve one or more data objects identified in the data object identification response. An access request may be based on a URL, such as an HTTP and/or FTP URL that identifies and locates a data object. Alternatively or additionally, an access request may be sent in a communication supported by a communications agent **403**, in a manner analogous to that illustrated with respect to FIG. 9A. See U. S. Patent Application No. 12/833,014 (published US 2012-0011207 A1) filed on 2010/07/09, entitled "Methods, Systems, and Program Products for Processing a Request for a Resource in a Communication" and Application No. 12/833,016 (published US 2012-0011444 A1) filed on 2010/07/09, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR REFERENCING AN ATTACHMENT IN A COMMUNICATION". As described above, content manager component **409** in execution environment **401** of first node **502** may package the one or more data representations including a representation of an access request into a message in a communication. The message is formatted according to a communications protocol supported by communications agent **403** in execution environment **401** of first node **502**. Communications protocol component **407** operating in execution environment **401** of first node **502** may send the message according to the specification(s) of the communications protocol. Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in a communication.

[0151] Content manager component **409** operating in execution environment **401** included in and/or otherwise provided by execution environment **401** of first node **502** may provide the packaged, encoded, and/or transformed data to communications

protocol component **407** via a com-out component **419**. Com-out component **419** as described above operatively couples communications agent **409** to communications protocol component **407** according to an interface provided by communications protocol component **407** for sending data in a communication according to a communications protocol. Communications protocol component **407** may further package and/or otherwise transform the data to send via network stack **405** for delivery via network **506** to execution environment **401** of second node **504**.

[0152] Returning to FIG. **2A**, a block **210** illustrates that the method yet further includes receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. Accordingly, a system for processing a data object identification request in a communication includes means for receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. For example, the arrangement in FIG. 3, includes content director component **310** that is operable for receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. FIG. **4** illustrates content director component **410** as an adaptation and/or analog of content director component **310** in FIG. 3. One or more content director components **410** operate in an execution environment **401**. The system for content director includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, via the network by the first execution environment and in response to sending the third message, at least a portion of the data object. In FIG. **4**, a content director component **410** is illustrated as a component of communications agent **403**.

[0153] FIG. **7** includes dataflow message **722** that illustrates a fourth message that includes an access response received via network **506** by execution environment **401** of first node **502** from execution environment **401** of second node **504**.

[0154] As described above, content manager component **409** operating in execution environment **401** of first node **502** may interoperate with communications protocol layer component **407** and/or network stack **405** to receive data in a communication, via network **506**, with another communications agent in another node. Content manager

component **409** is operatively coupled, via com-in component **411**, to communications protocol component **407** to receive the data from the other node. Thus, execution environment **401** of first node **502** may receive a message in a communication with first node **504** via a com-in component **411**. The message includes an access response to an access request. Content director component **410** may be included in content manager component **409** to detect one or more portions of a received message, such as a portion that is a response to an access request. Content director component **410** may invoke one or more content handler components **413** to process one or more data objects provided in the message according to their type.

[0155] In one aspect, a data object returned in a message in response to access request may be processed as an attachment and/or in a manner analogous to processing of an attachment for a particular communications type.

[0156] With reference to FIG. **2B**, a block **212** illustrates that the method includes receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. Accordingly, a system for processing a data object identification request in a communication includes means for receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. For example, the arrangement in FIG. **3B**, includes content manager component **309** that is operable for receiving, via a network by a second communications agent representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. FIG. **4** illustrates content manager component **409** as an adaptation and/or analog of content manager component **309** in FIG. **3B**. One or more content manager components **409** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for receiving, via a network by a second communications agent

representing a second user and operating in a second execution environment, a first message from a first communications agent representing a first user and operating in a first execution environment. In FIG. **4**, a content manager component **409** is illustrated as a component of communications agent **403**.

[0157] As described above, FIG. **7** illustrates first message **706**, including a data object identification request, received via network **506** by execution environment **401** of second node **504** from execution environment **401** of first node **502.**

[0158] In addition to packaging representations of data in a structure and/or format for sending a message according to a communications protocol, a content manager component **409** may operate to detect a data object identification request received and/or otherwise identified in a message received in a communication. Content manager component **409** operating in execution environment **401** of second node **504** may receive the message, illustrated by first message **706**, in a communication with execution environment **401** of first node **502**. Com-in component **411** in execution environment **401** of second node **504** may receive the message via communication protocol component **407** and network stack **405**. First message **706** may be delivered to execution environment **401** of second node **504** via network **506** based on a communications address of a communicant represented by communications agent **403** in execution environment **401** of second node **504**. A communications agent may represent more than one communicant, which may be identified by different addresses.

[0159] The first message **706** in FIG. **7** may be received in one or more packets via network **506** by network stack **405** and communications protocol component **407** in an instance of and/or analog of execution environment **401** including and/or otherwise provided by execution environment **401** of second node **504**. The data in the communication may be received by com-in component **411**. Com-in component **411** may provide the data to content manager component **409**. Content manager component **409** may determine one or more content types of the data. The content and/or portions of the content may be provided to one or more content type handler components **413** based on the one or more content types identified by content manager component **409**. For example, FIG. **8A** illustrates message portion **812a** including “text/plain” MIME type identifier **814a** as a content type identifier. Message portion **812a** may be provided to a

text/plain content handler **413**. Audio data in a voice communication may be provided to an audio content handler component **413**, and video data in a video communication may be provided to a video content handler component **413**.

[0160] Returning to FIG. **2B**, a block **214** illustrates that the method further includes detecting, by the second communications agent, a data object identification request included in the first message. Accordingly, a system for processing a data object identification request in a communication includes means for detecting, by the second communications agent, a data object identification request included in the first message. For example, the arrangement in FIG. **3B**, includes query handler component **302** that is operable for detecting, by the second communications agent, a data object identification request included in the first message. FIG. **4** illustrates query handler component **402** as an adaptation and/or analog of query handler component **302** in FIG. **3B**. One or more query handler components **402** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for detecting, by the second communications agent, a data object identification request included in the first message. In FIG. **4**, query handler component **402** is illustrated as a component of communications agent **403**.

[0161] FIG. **7** includes a dataflow message **708** that illustrates a detect data object identification request dataflow that may be internal to execution environment **401** of second node **504** and/or may include interoperation with another node via a network. With respect to FIG. **4**, dataflow message **708** may correspond to a data exchange and/or other interoperation between content handler component **409** and query handler component **402** to detect and/or otherwise process the data object identification request received in first message **706** by execution environment **401** of second node **504**. FIG. **7** further includes a dataflow message **710** that illustrates a search message for locating one or more data objects based on the data object identification request. The data object identification request may include and/or otherwise identify a data object matching criterion identified by data object information identified by the communicant represented by execution environment **401** of first node **502**. The one or more data

objects may be located and/or otherwise identified by determining a data object or data objects that match the data object matching criterion. The data object(s) may be in a data store accessible to execution environment **401** of second node **504**, such as one or more files in a file system of execution environment **401** of second node **504**. Dataflow message **710** may be internal to execution environment **401** of second node **504** and/or may be a message sent to a remote device, such as database server and/or a network attached storage device providing storage locations for some or all of a file system of execution environment **401** of second node **504**.

[0162] Com-in component **411** in execution environment **401** of second node **504**, as described above, provides the message, received from execution environment **401** of first node **502**, to content manager component **409**. Content manager component **409** is operatively coupled, via com-in component **411**, to communications protocol component **407** for receiving data in communications with other nodes, such as execution environment **401** of first node **502**. Thus, execution environment **401** of second node **504** may receive a message via a com-in component **411** in a communication with execution environment **401** of first node **502**. The message may include a data object identification request based on data object information identified by a user to communications agent **403** in execution environment **401** of first node **502**.

[0163] Data received in a communication, which may for example include a voice session and/or an instant message, may include data of various content types. In FIG. **4**, communications agent **403** in execution environment **401** of second node **504** includes one or more content handler components **413** to process data received according to its content type. Content manager component **409** may detect and/or otherwise identify a portion of the message as the data object identification request sent from execution environment **401** of first node **502**. The data object identification request may be identified by content manager component **409** according to a schema for the message and/or a schema for the data object identification request. For example, a portion of the message may include a MIME-type identifier that identifies the portion as a data object identification request and/or otherwise identifies the portion for routing to a query handler component **402** operating in and/or otherwise on behalf of execution environment **401** of second node **504**. Alternatively or additionally, the data object

identification request may have structure and/or content that is valid and identifiable according to a schema for defining and/or otherwise identifying valid data object identification requests. A query handler component, may be provided as a type of content handler component, in an aspect. Alternatively or additionally, a query handler component may interoperate with one or more content handler components to detect a data object identification request in a message received by a content manager component.

[0164] As such, query handler component **402** may operate along with other content handler components **413** to process data representations received in the message **706** from execution environment **401** of first node **502**. Query handler component **402** may also provide transformed data from the representations to one or more user interface element handler components **415**. Query handler component **402** operates to validate and process data object identification requests according to a schema defining valid data object identification requests. Various portions of a message including a data object identification request may be provided to one or more content handler components **413**, including query handler component **402**, to interoperate with presentation controller component **417** in execution environment **401** of second node **504** to present some or all of the received message, including the data object identification request.

[0165] FIG. **6D** illustrates a window, request window UI element **602d**, presenting an exemplary representation of a message received from execution environment **401** of first node **502** in a communication with second node **504**. As in FIG. **6A**, request window UI element **602d** includes a contactor UI element **604d**, a contactee UI element **606d**, a presentation space **608d** of the request window UI element **602d** that includes a user message UI element **610d** including a text portion of the message provided by the communicant represented by execution environment **401** of first node **502**. The various UI elements may be presented by one or more UI element handler components **415** in communications agent **403** operating in execution environment **401** of second node **504**. A reply UI element **648d** and a reply all UI element **648d** are illustrated as exemplary UI elements in presentation space **608d** presented to allow a user of execution environment **401** of second node **504** to respond and/or otherwise perform an

operation based on the received message. FIG. **6D** illustrates a data object identification request UI interface element **652d** where the data object identification request identified in the message from execution environment **401** of first node **502** is represented to the communicant represented by the execution environment **401** of second node **504**. In an aspect, the communicant is allowed to authorize the request by providing an input corresponding to a search and reply UI element **654d**. In another aspect, the communicant may be allowed to modify the received data object identification request.

[0166] FIG. **6E**, illustrates a reply window UI element **602e** that, in an aspect, may be presented by communications agent **403** in execution environment **401** of second node **504** in response to an user input corresponding to reply UI element **648d** detected by a corresponding UI element handler component **415**. In an aspect, a user of communications agent **403** operating in in execution environment **401** of second node **504** may provide input authorizing some or all of the processing of a data object identification request. Reply window UI element **602e** in FIG. **6E** includes a search UI element **656e** for receiving user input authorizing and/or otherwise instructing execution environment **401** of second node **504** to locate data objects that match the data object identification request, illustrated in a data object identification requested UI element **658e**.

[0167] As described herein in an aspect, content manager component **409** may detect content type information to detect a data object identification request in a communication. For example, the message portion illustrated in FIG. **8A** may be received in first message **706**. Content manager component **409** may detect "application/id-request" MIME type identifier in content-type header **804a**. The "application/id-request" MIME type identifier may be defined to identify a data object identification request based on one or more matching criteria for identifying a requested data object. Content manager component **409** may identify data object identification request portion **802a** as including the data object identification request.

[0168] In response to detecting a data object identification request in first message **706**, content manager component **409** may provide some or all of the data object identification request to query handler component **402**. For example, query handler

component **402** may be configured to operate according to a schema defining a format and/or a vocabulary for an XML-based language for id-request documents. Content manager component **409** may provide id-request document **806a**, as a data object identification request, to query handler component **402**. Query handler component **402** may operate according to the id-request schema. In an aspect, a query handler component may process more than one data object identification request content type. Alternatively or additionally, execution environment **401** may include multiple query handler components **402** for supporting multiple data object identification request content types.

[0169] Returning to FIG. **2B**, a block **216** illustrates that the method yet further includes generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. Accordingly, a system for processing a data object identification request in a communication includes means for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. For example, the arrangement in FIG. **3B**, includes query director component **312** that is operable for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request. FIG. **4** illustrates query director component 412 as an adaptation and/or analog of query director component **312** in FIG. **3B**. One or more query director components **412** operate in an execution environment **401**. The system for query director includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for generating, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is

identified by processing the data object identification request. In FIG. **4**, a query director component **412** is illustrated as a component of application **403**.

[0170] FIG. **7** includes dataflow message **712** that illustrates a create data object identification response data flow to process results returned from search dataflow **710**. In FIG. **4**, dataflow message **712** in FIG. **7** may represent data exchanged between query handler component **402** and query director component **412** to process results returned from a search operation performed based on the data object identification request, as illustrated by dataflow message **710**. Processing the results may include creating and/or otherwise constructing a data object identification response to the data object identification request identified in first message **706**. Dataflow message **712** may be internal to execution environment **401** of second node **504** and/or may include interoperation by execution environment **401** of second node **504** via a network with another node.

[0171] FIG. **6F**, illustrates an identify in response dialog UI element **660f** that, in an aspect, may be presented by communications agent **403** in execution environment **401** of second node **504** in response to processing the data object identification request to locate matching data objects. In an aspect, a user of communications agent **403** operating in execution environment **401** of second node **504** may provide input corresponding to search UI element **656e** in reply window UI element **602e** in FIG. **6E**. A corresponding UI element handler component **415** may operate to invoke query director component **412** to perform and/or otherwise to provide for performing one or more operations to locate one or more data objects based on the data object identification request. Query director component **412** may interoperate with one or more UI element handlers **415**, directly or indirectly via query handler component **402**, in an aspect, to present the UI element **660f** and its contents. Identify in response dialog UI element **660f** includes results pane UI element **662f** including representations of two data objects located and/or otherwise identified by query director component **412** based on the data object identification request. A user may select one or more data objects to identify in a data object identification response. An include UI element **664f** may be presented, as illustrated, to allow a user to instruct communications agent **403** to identify one or more selected data objects in results pane UI element **662f** in a data

object identification response. An include all UI element **666f** is illustrated to allow for user input instructing communications agent **403** to identify all data objects represented in the UI element **660f** in a data object identification response in a communication with execution environment **401** of first node **502**. A cancel UI element **668f** allows a user to end presentation of dialog UI element **660f**. A user may opt to identify no data objects represented in the dialog UI element **660f** and located and/or otherwise identified by query director component **412** based on the data object identification request.

[0172] Data objects located and/or otherwise identified by query director component **412** may be identified to IRB component **414** to construct and/or otherwise create a data object identification response identifying the data objects. As described above, in an aspect, a user may be allowed to select which data objects, if any, are to be identified in a data object identification response from one or more data objects located and/or otherwise identified by query director component **412** based on the data object identification response. In another aspect, communications agent **403** in FIG. **4** may invoke query director component **412** automatically to locate one or more data objects based on a data object matching criterion identified in a data object identification request and in response to detecting a data object identification request in a communication. In another aspect, communications agent **403** may invoke query director component **412** automatically to locate one or more data objects based on a data object identification request in a communication in response to detecting an indication to present, via a output device, some or all of a user message received in a message in the communication. Query director component **412** may be invoked and/or may operate based on the data object identification request and/or data object matching criterion identified to locate a data object prior to, during, and/or after other content received in the communication is processed.

[0173] The method illustrated in FIG. **2B** may include additional aspects supported by various adaptations and/or analogs of the arrangement of components in FIG. **4**. As described above, in one aspect a data object may be located and/or otherwise identified automatically in response to detecting the data object identification request. In another aspect, locating a data object may include presenting a locator UI element representing a data object identification request. Search UI element **656e** in FIG. **6E** exemplifies one

type of locator UI control. Search UI element **656e** may be presented by presentation controller component **417** as described above. A user input corresponding to a search UI element, such as search UI element **656e**, may be detected by presentation controller component **417**. Presentation controller component **417** may instruct query director component **412** to locate the data object based on a data object identification request represented by the data object identification request UI element **658e**.

[0174] In another aspect, a data object identification request UI element may be presented to receive one or more additional matching criteria from a user. Based on the data object identification request UI element and input from the user, query director component **412** may receive one or more additional data object matching criterion information and/or may receive input to modify a data object matching criterion identified in a received data object identification request. Query director component **412** may locate one or more data objects in response to receiving the additional criterion.

[0175] Returning to FIG. **2B**, a block **218** illustrates that the method yet further includes sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. Accordingly, a system for processing a data object identification request in a communication includes means for sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. For example, the arrangement in FIG. **3B**, includes identification response builder component **314** that is operable for sending a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. FIG. **4** illustrates identification response builder component **414** as an adaptation and/or analog of identification response builder component **314** in FIG. **3B**. One or more identification response builder components **414** operate in an execution environment **401**. The system for processing a data object identification request in a communication includes one or more processors and logic encoded in one or more tangible media for execution by the one or more processors that when executed is operable for sending a

second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object. In FIG. **4**, a identification response builder component **414** is illustrated as a component of application **403**.

[0176] As described above, FIG. **7** includes second message dataflow message **714** illustrating second message sent via network **506** to execution environment **401** of first node **502** from execution environment **401** of second node **504**. Second message **714** includes a data object identification response in response to the data object identification request from first node **502** received and/or otherwise identified in first message **706**.

[0177] IRB component **414** may provide the data object identification response to content manager component **409** to include in a message, illustrated by second message **714**. As described above, content manager component **409** may receive and/or otherwise identify various portions of a message, including a data object identification response, from one or more content handler components **413**. The data object identification response may be provided to content manager component **409**. As described above, content manager component **409** in execution environment **401** of second node **504** may package the one or more data representations including a representation of the data object identification response into the message formatted according to a suitable communications protocol. Communications protocol component **407** operating in execution environment **401** of second node **504** may send the data as a message in a communication with execution environment **401** of first node **502** according to the specification(s) of the communications protocol. Content manager component **409** may alternatively or additionally encode and/or otherwise transform one or more of the data representations for sending in a data stream such as voice stream and/or a video stream for communicating in the communication to the communications agent.

[0178] FIG. **8B** illustrates an exemplary data object identification response described in detail above. Second message portion **802b** illustrates a user message portion that may be included in second message **714**. Note that a user message portion may be empty or absent in some aspects. Data object identification response portion **804b** illustrates

the data object identification response as a list of URIs identifying data objects in a file system of execution environment **401** of second node **504**. User Message portion **802b** and data object identification response portion **804b** may be provided by respective content handler components **413** to content manager component **409**. Content manager component **409** may construct the content as illustrated in FIG. **8B** and/or otherwise provide for encoding, translating, combining, and/or otherwise preparing the content for sending in second message **714**.

[0179] The methods illustrated in FIG. **2A-B** may include additional aspects supported by various adaptations and/or analogs of the arrangement of components in FIG. **3A-B**. In various aspects, a data object identification request may be based on a date, a time, a length of time, a file type, a database record key, content of the data object, a content type identifier, a format rule, a vocabulary, a role of a user, a security attribute, a location in a data store such as a file system, an attribute of an identified data object, a size, a task, a transaction, a state, a user, a group, a requester, a relationship including a requesting user and a responding user, a keyword, a tag, a folder, and/or a path portion of a data object identifier – to name a few examples.

[0180] In another aspect, a data object located and/or otherwise identified in response to receiving a data object identification request must meet match a query identified by the data object identification request. Alternatively or additionally, a data object identification request may include an instruction and/or input for generating a data object. Locating the data object may include generating the data object. Generating a data object may include creating the data object and/or may include modifying and/or otherwise transforming an existing data object. For example, execution environment **401** of second node **504** may include a template stored in a file system. A data object identification request may identify the template. A query handler component **402** may operate to identify a document or other data object that may be created, is being created, and/or has been created based on the template. In response to an access request, execution environment **401** of second node **504** may return the document.

[0181] As described above, in various aspects, a data object identification request may be detected and/or represented based on various syntaxes, grammars, vocabularies, and/or languages. For example, a data object identification request may be identified

and/or represented according to a file system search syntax, a regular expression language, a structured query language (SQL) query, a universal data object identifier schema, an XPATH based language, an XQuery based language, an XML based language, an HTML based language (form-based), and/or a keyword-value pair based language.

[0182] A data object identification request in a communication may be communicated via a network according to a first communications protocol. A user message may be exchanged between communicants in the communication via a second communications protocol. For example, a data object identification request may be sent on behalf of a first communicant by execution environment **401** of first node **502** including a communications agent **403** representing the first communicant. The data object identification request may be included in a communication along with a text message sent to the execution environment **401** of second node **504**. The communicants represented by the execution environment **401** of first node **502** and the execution environment **401** of second node **504** may communicate via one or more exchanges of audio exchanged via, for example, a voice over IP (VoIP) communications protocol.

[0183] Exemplary data objects that may be requested via a data object identification request include a file, a program component, a data base record, video data, audio data, markup language, binary data, text data, an output of a service. Requested data objects may be pre-existing, volatile, and/or generated in response to the request.

[0184] As described above, receiving data object information may include receiving, via network from a node that received the data object information from the user, a message that identifies the data object information. For example, data object information may be received from a browser by a web server.

[0185] Data object information and/or a data object identification request may be generated, detected, and/or otherwise processed according to a schema that identifies at least one of a rule and a vocabulary that defines a valid data object identification request. Similarly, a data object identification response, an access request, and a response to an access request may be generated, detected, and/or otherwise processed according to a schema. One or more schemas may alone or together define one or more of valid data object information, a data object identification request, data

object identification response, an access request, and/or an access response. A schema, as just described, may be identified based on data received and/or otherwise identified by a node hosting a communications agent via the network from another node hosting a communications agent and/or from a path node included in a network path in the network communicatively coupling the node to one or more nodes that respectively host communications agents.

[0186] In a further aspect, multiple alternative schemas may be defined for each of data object information, data object identification requests, data object identification responses, access requests, an/or an access responses processing data object identification requests. A schema may be selected and/or otherwise identified from the multiple schemas by execution environment **401** of first node **502** and the execution environment **401** of second node **504** and/or based on information exchanged by the two nodes. For example, a schema may be selected by the first communicant and/or the second communicant. One may inform the other, via a communications protocol and/or via any other suitable means. The communicants may negotiate which schema to use via any suitable means.

[0187] The communications agents in execution environment **401** of second node **504** and execution environment **401** of first node **502** may select a schema For example, a schema may be identified and/or otherwise selected based on a communications protocol supported and/or a communications protocol not supported by one or both nodes. That is, in negotiating which communication protocol to use allowing communicants to exchange information, a schema may be identified. A schema may be identified and/or otherwise selected based on an attribute of the data object. A schema for identifying image data may be different than a schema for editable documents. A schema for identifying a data object in a first file system may be different than a schema for identifying a data object in a different file system.

[0188] As described above, data object information and/or a data object identification request may identify a scope that specifies all or some portion of one or more data stores for locating a data object. For example, A communications agent **403** in execution environment **401** of first node **502** may receive user input that identifies a

folder in a file system of the execution environment **401** of second node **504** that defines a scope for locating a data object.

[0189] Data object information and/or a data object identification request may be represented in a representation that includes a complete or a portion of at least one of a keyword expression, a regular expression, an XQuery expression, an XPath expression, a file system path expression, and a structured query language statement.

[0190] Further as also described above, a data object identification request, a data object identification response, an access request, an/or an access response may be included in a message according to a schema that identifies the message as valid for a specified communications protocol. Those skilled in the art are aware that schemas for email differ from schemas for instant messages and from streamed audio, for example.

[0191] A data object identification request, a data object identification response, an access request, and/or an access response may be exchanged in a communication in which audio data and/or image data is also exchanged. Multimedia may be a packaged to include one or more data streams which may include text based data streams.

[0192] A data object identification request, a data object identification response, an access request, and/or an access response may be included in an email, an instant message, a multi-media message, a short-message service message, and/or a data stream. The data stream may include audio data, image data, and/or text data.

[0193] A data object identification request, a data object identification response, an access request, an/or an access response may be included in and/or detected in a communication based on a location in the communication and/or based on a marker, such as an identifier in the communication that identifies a location for the data object identification request, the data object identification response, the access request, and/or the access response. The location may be absolute, such as fixed number of bytes or fields from the start or end of a message. Alternatively or additionally, the location may be identified by a location relative to another detectable portion of the communication. FIGs. **8A-B** illustrate the use of MIME type identifiers as markers that identify locations for one or more of a data object identification request, a data object identification response, an access request, an/or an access response in a communication.

[0194] A data object identification request, a data object identification response, an access request, an/or an access response or respective portions thereof may include at least a portion of a keyword expression, a regular expression, expression including a Boolean operator, an expression including a precedence information, and a structured query language statement.

[0195] With respect to the method illustrated in FIG. **2B**, the method as described above may include locating and/or otherwise identifying a data object automatically in response to receiving an access request that matches and/or otherwise identifies the data object. Further, a data object identification response may be created and/or otherwise generated automatically, in response to locating and/or otherwise identifying the data object.

[0196] In an aspect, in response to receiving and/or otherwise detecting a data object identification request, a user interface may be presented based on the data object identification request to prompt a user to process the data object identification request in some manner, such as authorizing the request and/or modifying the request by restricting its scope. In response to detecting user input targeting and/or otherwise corresponding to the presented user interface, one or more data objects may be located and/or otherwise identified based on the data object identification request.

[0197] A user may change a received data object identification request. One or more data objects may be located and/or otherwise identified based on the changed data object identification request.

[0198] In a further aspect, once one or more data objects are located and/or otherwise identified based on a data object identification request, a user interface may be presented to allow a user to identify the one or more data objects. User information may be received that selects one or more data objects to identify in a data object identification response and/or to not identify in the data object identification response. A user, in an aspect, may be allowed to identify a data object not located and/or otherwise identified by the user's node. The data object, based on the user input, is identified in the data object identification response.

[0199] In light of the description provided above, a data object identification response may be generated automatically in response to detecting a corresponding data object

identification request by an execution environment **401** of first node **502**. Further, the automatically generated data object identification response, may be sent to the node that sent the corresponding data object identification request, automatically in response to the generating of the data object identification response.

[0200] With respect to the method illustrated in FIG. **2A**, the third message may be sent automatically in response to detecting the data object identification response. In another aspect, a representation of a received data object identification response may be presented, via an output device, to a user. The representation may identify one or more data objects located and/or otherwise identified based on the data object identification request. The representation may be presented automatically, in response to detecting the data object identification response. Further, a user input may be detected that corresponds to and/or otherwise targets the representation. The user input may be processed as an indication to retrieve the data object. As a result an access request may be created or otherwise generated that identifies the data object. The access request may be sent in a message via the network to retrieve the data object. The message may be sent to a node that sent the data object identification response and/or to another node. The third message may be sent by the communications agent in the execution environment **401** of first node **502** to the communications agent in the execution environment **401** of second node **504** via a communications protocol. The third message sent via the communication protocol may include a communications address, in an address space of the communications protocol, that identifies a user represented by the communications agent in the execution environment **401** of second node **504**. The third message may include a data object identification request, a data object identification response, and/or a data object sent in response to previously receiving an access response by the execution environment **401** of first node **502**. The third message may include an access request that identifies the data object. The access request may include a universal data object identifier (URI) that identifies the data object.

[0201] With respect to the method illustrated in FIG. **2B**, in response to receiving an access request, a data object identified in the request may be sent. The data object may be sent automatically, in response to receiving the access request. The data object may

be sent as attachment in a communication. The data object may be sent by a communications agent in the execution environment **401** of second node **504** to a communications agent in the execution environment **401** of first node **502** via a communications protocol. The data object may be sent via a communication that includes a communications address, in an address space of the communications protocol, that identifies the first communicant represented by the execution environment **401** of first node **502**. The communication that includes the data object may include a data object identification request, a data object identification response, and/or an access request sent in response to a data object identification response sent previously by the first node.

[0202] The methods illustrated in FIGs. **2A-B** may yet further include additional aspects supported by various adaptations and/or analogs of the arrangement of components in FIGs. **3A-B**, respectively.

[0203] Performing the method illustrated in FIG. **2A** and/or the method illustrated in FIG. **2B** and/or any of its extension and/or in any of its aspects may include one or more of calling a function or method of an object, sending a message via a network; sending a message via an inter-process communication mechanism such as a pipe, a semaphore, a shared data area, and/or a queue; and/or receiving a request such as poll and responding to invoke, and sending an asynchronous message.

[0204] Output devices suitable for presenting a representation of a data object and/or a tag include a visual output device, an audio output device, and a tactile output device. One output device may present a data object and another output device may present a tag with which the data object is tagged.

[0205] Any action included in performing the method illustrated in FIG. **2A** as well as any action included in in performing the method illustrated in FIG. **2B** may include sending and/or receiving a message via a network. Further, a message, included in performing any of the subject matter described herein and/or any of its extensions in any of its aspects, may an asynchronous message without a corresponding request.

[0206] To the accomplishment of the foregoing and related ends, the descriptions and annexed drawings set forth certain illustrative aspects and implementations of the disclosure. These are indicative of but a few of the various ways in which one or more

aspects of the disclosure may be employed. The other aspects, advantages, and novel features of the disclosure will become apparent from the detailed description included herein when considered in conjunction with the annexed drawings.

[0207] It should be understood that the various components illustrated in the various block diagrams represent logical components that are configured to perform the functionality described herein and may be implemented in software, hardware, or a combination of the two. Moreover, some or all of these logical components may be combined, some may be omitted altogether, and additional components may be added while still achieving the functionality described herein. Thus, the subject matter described herein may be embodied in many different variations, and all such variations are contemplated to be within the scope of what is claimed.

[0208] To facilitate an understanding of the subject matter described above, many aspects are described in terms of sequences of actions that may be performed by elements of a computer system. For example, it will be recognized that the various actions may be performed by specialized circuits or circuitry (e.g., discrete logic gates interconnected to perform a specialized function), by program instructions being executed by one or more processors, or by a combination of both. The description herein of any sequence of actions is not intended to imply that the specific order described for performing that sequence must be followed.

[0209] Moreover, the methods described herein may be embodied in executable instructions stored in a non-transitory computer readable medium for use by or in connection with an instruction execution machine, system, apparatus, or device, such as a computer-based or processor-containing machine, system, apparatus, or device. As used here, a "non-transitory computer readable medium" may include one or more of any suitable media for storing the executable instructions of a computer program in one or more forms including an electronic, magnetic, optical, and electromagnetic form, such that the instruction execution machine, system, apparatus, or device may read (or fetch) the instructions from the non-transitory computer readable medium and execute the instructions for carrying out the described methods. A non-exhaustive list of conventional exemplary non-transitory computer readable media includes a portable computer diskette; a random access memory (RAM); a read only memory (ROM); an

erasable programmable read only memory (EPROM or Flash memory); optical storage devices, including a portable compact disc (CD), a portable digital video disc (DVD), a high definition DVD (HD-DVD.TM.), and a Blu-ray.TM. disc; and the like

[0210] Thus, the subject matter described herein may be embodied in many different forms, and all such forms are contemplated to be within the scope of what is claimed. It will be understood that various details may be changed without departing from the scope of the claimed subject matter. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation, as the scope of protection sought is defined by the claims as set forth hereinafter together with any equivalents.

[0211] Thus, the subject matter described herein may be embodied in many different forms, and all such forms are contemplated to be within the scope of what is claimed. It will be understood that various details may be changed without departing from the scope of the claimed subject matter. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation, as the scope of protection sought is defined by the claims as set forth hereinafter together with any equivalents.

[0212] All methods described herein may be performed in any order unless otherwise indicated herein explicitly or by context. The use of the terms "a" and "an" and "the" and similar referents in the context of the foregoing description and in the context of the following claims are to be construed to include the singular and the plural, unless otherwise indicated herein explicitly or clearly contradicted by context. The foregoing description is not to be interpreted as indicating that any non-claimed element is essential to the practice of the subject matter as claimed.

I CLAIM:

1. A computer-implemented method, comprising:

displaying, at a first node, at least one first interface with a first user interface element and a second user interface element, utilizing first hypertext markup language-equipped code that is received via at least one network;

receiving, utilizing the at least one first interface at the first node, a file description via the first user interface element that includes a textbox;

sending, from the first node via the at least one network, the file description;

receiving, utilizing the at least one first interface at the first node, an indication of at least one of a date or a time via the second user interface element;

sending, from the first node via the at least one network, the indication of the at least one of the date or the time;

displaying, at the first node, at least one second interface with a third user interface element and a fourth user interface element, utilizing second hypertext markup language-equipped code that is received via the at least one network;

receiving, utilizing the at least one second interface at the first node, indicia associated with at least one email address via the third user interface element;

sending, from the first node via the at least one network, the indicia associated with the at least one email address;

detecting, at the first node, a selection of the fourth user interface element;

sending, from the first node via the at least one network, an indication of the selection of the fourth user interface element;

after sending the indication of the selection of the fourth user interface element and based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; receiving, at a second node via the at least one network, a first message indicating the at least one of the date or the time, and including a fifth user interface element and at least one sentence including the file description;

detecting, at the second node, a selection of the fifth user interface element of the first message;

sending, from the second node via the at least one network, an indication of the selection of the fifth user interface element of the first message;

after sending the indication of the selection of the fifth user interface element, displaying, at the second node, at least one third interface with a sixth user interface element and a seventh user interface element, utilizing third hypertext markup language-equipped code that is received via at least one network;

receiving, utilizing the at least one third interface at the second node, a selection of at least one file utilizing the sixth user interface element;

sending, from the second node via the at least one network, an indication of the at least one file;

detecting, utilizing the at least one third interface at the second node, a selection of the seventh user interface element;

sending, from the second node via the at least one network, an indication of the selection of the seventh user interface element;

after sending the indication of the selection of the seventh user interface element, receiving, at the first node, a second message that includes an eighth user interface element, and that does not include a file attachment with the second message;

detecting, at the first node, a selection of the eighth user interface element of the second message;

sending, from the first node via the at least one network, an indication of the selection of the eighth user interface element of the second message;

after sending the indication of a selection of the eighth user interface element of the second message, displaying, at the first node, at least one fourth interface with a reference to the at least one file;

detecting, utilizing the at least one fourth interface at the first node, a selection of the reference to the at least one file; and

providing access, at the first node, to the at least one file, utilizing the reference.

2. The method of claim 1, wherein the at least one first interface and the at least one second interface are separate windows displayed utilizing a single web page.

3. The method of claim 1, wherein the first message is received utilizing a simple mail transfer protocol (SMTP) and the second message is received utilizing a hypertext transfer protocol (HTTP).

4. The method of claim 1, wherein the first message and the second message are both received utilizing a simple mail transfer protocol (SMTP).

5. The method of claim 1, wherein the second message is at least partially pre-written and is automatically generated without user involvement after the indication of the selection of the seventh user interface element is sent.

6. The method of claim 1, wherein the at least one of the date or the time of the indication of the at least one of the date or the time is a criterion, and access is provided to the at least one file based on whether the criterion is met.

7. The method of claim 1, wherein the file description is capable of being received in response to a first single user input, and the indication of the at least one of the

date or the time is capable of being received in response to a second single user input that immediately follows the first single user input.

8. The method of claim 1, wherein the second message does not include the file attachment, for avoiding the at least one file from having to be communicated to and stored at the first node until the communication and the storage is initiated by a user of the first node.

9. The method of claim 1, wherein a first user of the first node is capable of requesting the at least one file and accessing the at least one file, by permitting a second user of the second node to store the at least one file at at least one server that is accessible by the first user of the first node, without requiring the at least one file to be communicated from the second node to the first node as a message attachment.

10. An apparatus, comprising:

a first node configured to:

display, at the first node, a first user interface element, a second user interface element, and a third user interface element;

receive, at the first node, text associated with one or more desired files via the first user interface element that includes a textbox;

send, from the first node via at least one network, the text associated with the one or more desired files;

receive, at the first node, an object associated with at least one email address via the second user interface element;

send, from the first node via the at least one network, the object associated with the at least one email address;

detect, at the first node, a selection of the third user interface element; and

send, from the first node via the at least one network, a message associated with the selection of the third user interface element;

said first node further configured to communicate with a second node that is configured to:

after the message associated with the selection of the third user interface element is sent and based on the text associated with the one or more desired files and the object associated with the at least one email address; receive, at the second node via the at least one network, a first message including a fourth user interface element and at least one sentence including the text associated with the one or more desired files;

detect, at the second node, a selection of the fourth user interface element of the first message;

send, from the second node via the at least one network, a message associated with the selection of the fourth user interface element of the first message;

after the message associated with the selection of the fourth user interface element is sent, display, at the second node, a fifth user interface element and a sixth user interface element;

detect, at the second node, a selection of at least one file;

send, from the second node via the at least one network, a message associated with the at least one file;

detect, at the second node, a selection of the sixth user interface element; and

send, from the second node via the at least one network, a message associated with the selection of the sixth user interface element;

said first node further configured to:

after the message associated with the selection of the sixth user interface element is sent, receive, at the first node, a second message that includes a seventh user interface element, and that does not include a file attachment with the second message;

detect, at the first node, a selection of the seventh user interface element of the second message;

send, from the first node via the at least one network, a message associated with the selection of the seventh user interface element of the second message;

after the message associated with the selection of the seventh user interface element of the second message is sent, display, at the first node, an eighth user interface element associated with a reference to the at least one file; and

provide access, at the first node, to the at least one file utilizing the reference.

11. The apparatus of claim 10, wherein the first node is configured such that:

the first message and the second message are emails;

the object is the at least one email address or an alias associated with the at least one email address;

the second message is at least partially pre-written and is automatically generated without user involvement after the message associated with the selection of the sixth user interface element is sent;

the at least one file is conditionally shared based on whether a date-related criterion or a time-related criterion is met;

the second message does not include the file attachment, for avoiding the at least one file from being communicated to and stored at the first node until the communication of the at least one file is initiated by a first user of the first node via at at least one server that stores the at least one file.

12. An apparatus, comprising:

at least one non-transitory memory storing instructions; and

one or more processors in communication with the at least one non-transitory memory, wherein the one or more processors execute the instructions to:

cause, at a first node, display of at least one first interface with a first user interface element and a second user interface element, utilizing first hypertext markup language-equipped code that is sent via at least one network;

receive, from the first node via the at least one network, a file description via the first user interface element that includes a textbox, utilizing the at least one first interface;

receive, from the first node via the at least one network, an indication of at least one of a date or a time via the second user interface element, utilizing the at least one first interface;

cause, at the first node, display of at least one second interface with a third user interface element and a fourth user interface element, utilizing second hypertext markup language-equipped code that is sent via the at least one network;

receive, from the first node via the at least one network, indicia associated with at least one email address via the third user interface element, utilizing the at least one second interface;

receive, from the first node via the at least one network, an indication of a selection of the fourth user interface element, utilizing the at least one second interface;

in response to the receipt of the indication of the selection of the fourth user interface element and based on the file description, the indication of the at least one of the date or the time, and the indicia associated with the at least one email address; generate a first message indicating the at least one of the date or the time, and including a fifth user interface element and at least one sentence including the file description;

send, to a second node via the at least one network, the first message;

receive, from the second node via the at least one network, an indication of a selection of the fifth user interface element of the first message;

in response to the receipt of the indication of the selection of the fifth user interface element, cause, at the second node, display of at least one third interface with a sixth user interface element and a seventh user interface element, utilizing third hypertext markup language-equipped code that is sent via at least one network;

receive, from the second node via the at least one network, an indication of a selection at least one file utilizing the sixth user interface element, utilizing the at least one third interface;

receive, from the second node via the at least one network, an indication of a selection of the seventh user interface element, utilizing the at least one third interface;

in response to the receipt of the indication of the selection of the seventh user interface element, generate a second message that includes an eighth user interface element, and that does not include a file attachment with the second message;

send, to the first node via the at least one network, the second message;

receive, from the first node via the at least one network, an indication of a selection of the eighth user interface element of the second message;

in response to the receipt of the indication of the selection of the eighth user interface element of the second message, cause, at the first node, display of at least one fourth interface with a reference to the at least one file for allowing access to the at least one file at the first node.

13. The apparatus of claim 12, wherein the apparatus is configured such that the indicia associated with the at least one email address includes an alias associated with the at least one email address.

14. The apparatus of claim 12, wherein the apparatus is configured such that the indicia associated with the at least one email address includes the at least one email address.

15. The apparatus of claim 12, wherein the apparatus is configured such that the at least one first interface and the at least one second interface are displayed utilizing a single web page.

16. The apparatus of claim 12, wherein the apparatus is configured such that the at least one first interface and the at least one second interface are separate windows displayed utilizing a single web page.

17. The apparatus of claim 12, wherein the apparatus is configured such that the first message is sent utilizing a simple mail transfer protocol (SMTP) and the second message is sent utilizing a hypertext transfer protocol (HTTP).

18. The apparatus of claim 12, wherein the apparatus is configured such that the first message and the second message are both sent utilizing a simple mail transfer protocol (SMTP).

19. The apparatus of claim 12, wherein the apparatus is configured such that the second message is at least partially pre-written and is automatically generated without user involvement in response to the receipt of the indication of the selection of the seventh user interface element.

20. The apparatus of claim 12, wherein the apparatus is configured such that the at least one of the date or the time of the indication of the at least one of the date or the time is a criterion, and the indication of the at least one file is received based on whether the criterion is met.

ABSTRACT

Methods, systems, and apparatuses are provided that are configured to: display, at a first node, at least one first interface with a first user interface element and a second user interface element; display, at the first node, at least one second interface; generate a first message; display, at the second node, at least one third interface; generate a second message that does not include a file attachment with the second message; display, at the first node, at least one fourth interface with a reference to at least one file; and allow access, at the first node, to the at least one file.

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION
First Named Inventor/Applicant Name:	Robert Paul Morris
Filer:	Patrick Edgar Caldwell
Attorney Docket Number:	DR0149D-CIP

Filed as Small Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
UTILITY FILING FEE (ELECTRONIC FILING)	4011	1	75	75
UTILITY SEARCH FEE	2111	1	330	330
UTILITY EXAMINATION FEE	2311	1	380	380
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
			Total in USD ($)	**785**

Electronic Acknowledgement Receipt

EFS ID:	38718432
Application Number:	16803980
International Application Number:	
Confirmation Number:	8437
Title of Invention:	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION
First Named Inventor/Applicant Name:	Robert Paul Morris
Customer Number:	92045
Filer:	Patrick Edgar Caldwell
Filer Authorized By:	
Attorney Docket Number:	DR0149D-CIP
Receipt Date:	27-FEB-2020
Filing Date:	
Time Stamp:	23:09:17
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$785
RAM confirmation Number	E20202QN09426887
Deposit Account	
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Oath or Declaration filed	DR0149_Declaration_SIGNED_vF_27-Feb-2020.PDF	150675 8668cc715124c840bbc8b25a37abe364b014e6d0	no	2
Warnings:					
Information:					
2	Drawings-only black and white line drawings	DR0149_Drawings_vF_27-Feb-2020.pdf	1712012 71b471fa2d6723549cfd8ecff49f90f20ff102b2	no	15
Warnings:					
Information:					
3	Power of Attorney	DR0149_POA_MotionOffense_vF_27-Feb-2020.pdf	163126 68e2d3f0e8e16335799fabebb03ed1c847c8b4c4	no	1
Warnings:					
Information:					
4	Application Data Sheet	DR0149D-CIP_ADS_vF_27-Feb-2020.pdf	1873938 4322cde4d5dc20cf5d7fe693929209e534d7b3e2	no	9
Warnings:					
Information:					
5	Information Disclosure Statement (IDS) Form (SB08)	DR0149D-CIP_IDS_20200227_vF.pdf	1057625 16e76a7deda8864e47abf5b53b0e40d7574470f8	no	18
Warnings:					
Information:					
6		DR0149D-CIP_Spec_vF_27-Feb-2020.pdf	423112 bfcca020353e27ae9366f839bf89dc709dc1c27b	yes	91
	Multipart Description/PDF files in .zip description				
	Document Description		**Start**	**End**	

	Specification		1		74
	Claims		75		90
	Abstract		91		91
Warnings:					
Information:					
7	Fee Worksheet (SB06)	fee-info.pdf	35507 d185fcddd8ca21d315fe86d57844f0e381ed434c	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	5415995		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/AIA/01 (06-12)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A DATA OBJECT IDENTIFICATION REQUEST IN A COMMUNICATION

As the below named inventor, I hereby declare that:

This declaration is directed to:

- [■] The attached application, or
- [ ] United States application or PCT international application number _____

filed on _____.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Robert Paul Morris Date (Optional) :_____

Signature: /Robert Paul Morris/

Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form or must have been previously filed. Use an additional PTO/AIA/01 form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 1 minute to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

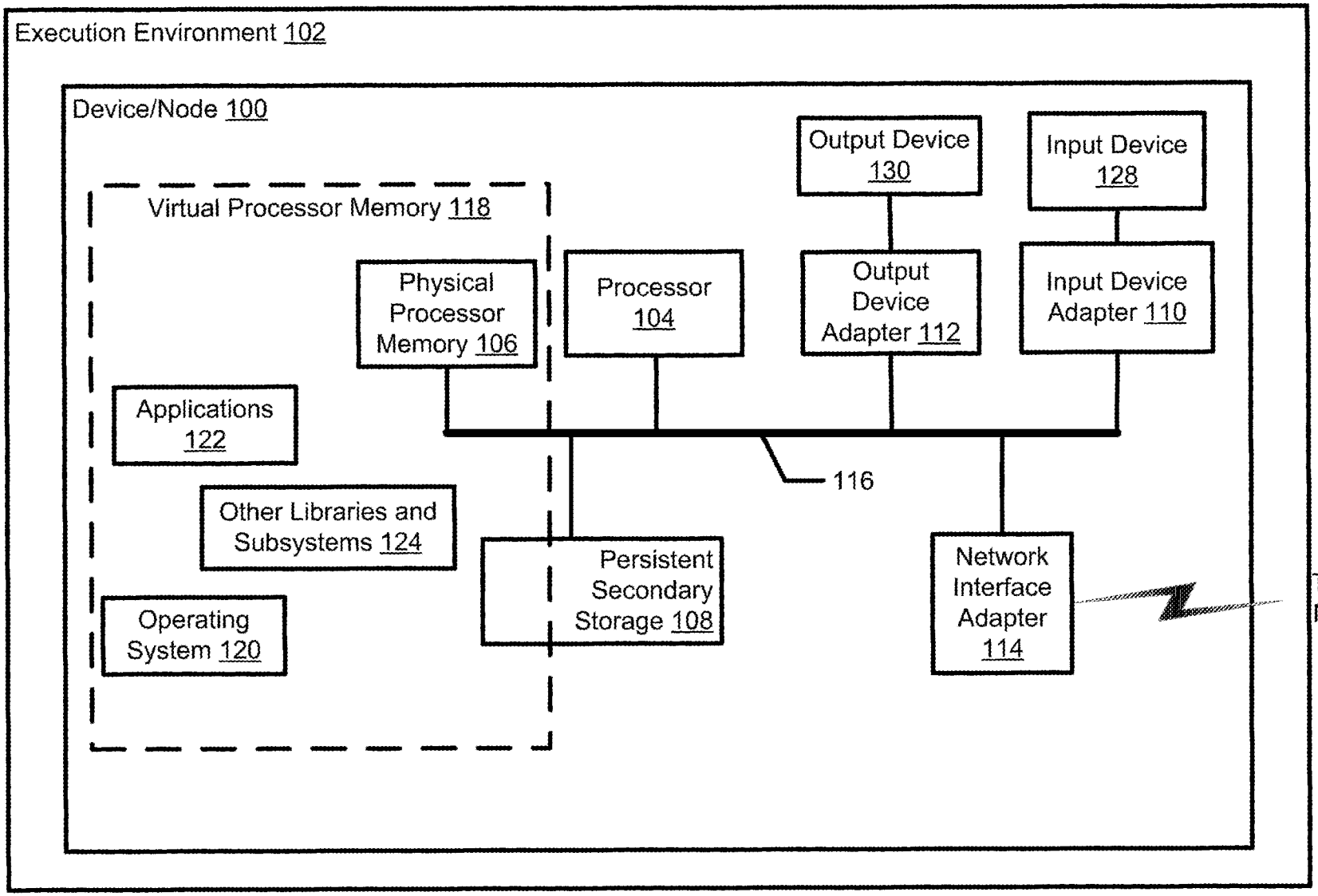

Execution Environment 102
Device/Node 100
Virtual Processor Memory 118
Physical Processor Memory 106
Processor 104
Output Device 130
Input Device 128
Output Device Adapter 112
Input Device Adapter 110
Applications 122
Other Libraries and Subsystems 124
Operating System 120
Persistent Secondary Storage 108
116
Network Interface Adapter 114
To/From Network

FIG. 1

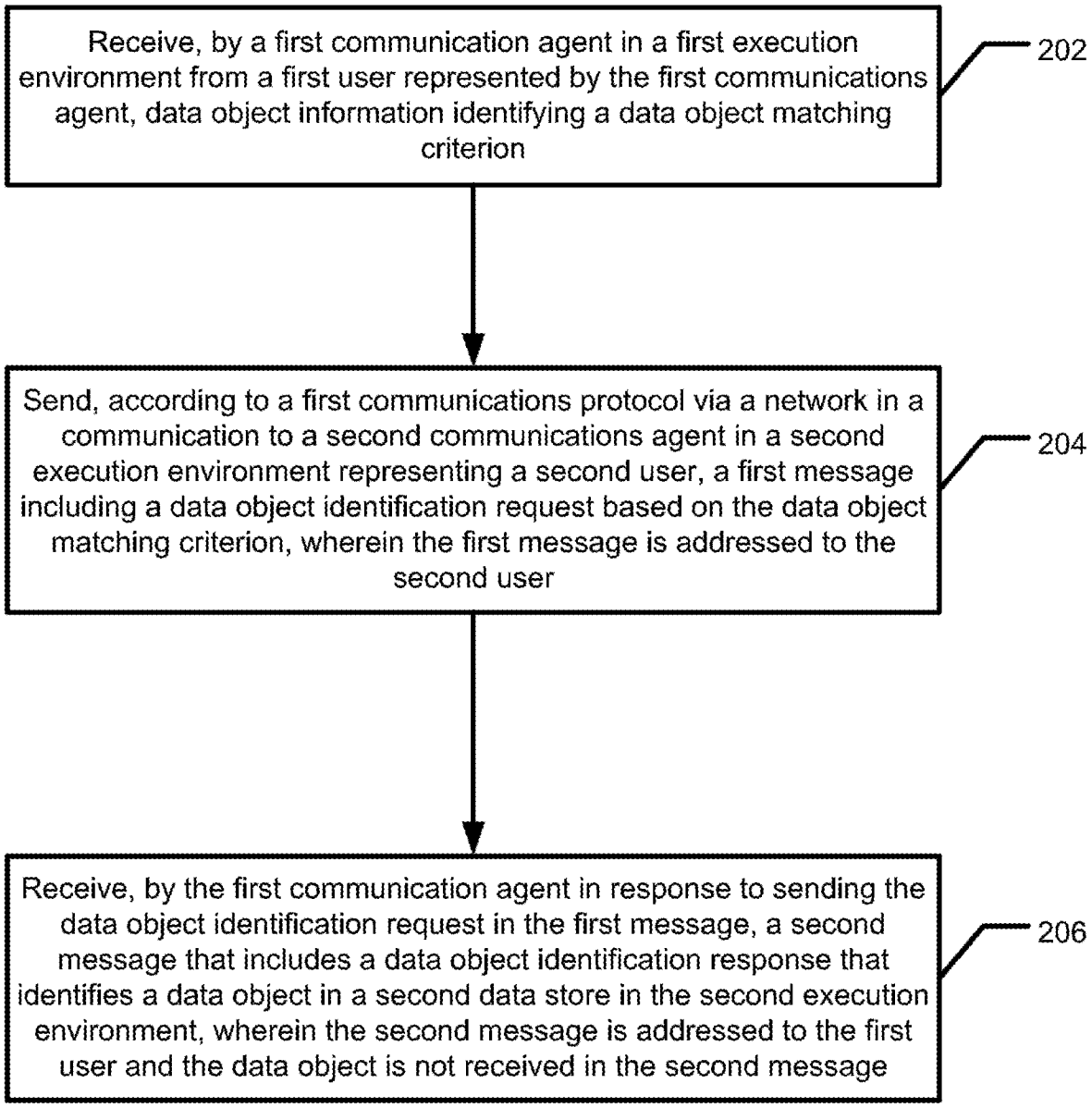

Receive, by a first communication agent in a first execution environment from a first user represented by the first communications agent, data object information identifying a data object matching criterion
202
Send, according to a first communications protocol via a network in a communication to a second communications agent in a second execution environment representing a second user, a first message including a data object identification request based on the data object matching criterion, wherein the first message is addressed to the second user
204
Receive, by the first communication agent in response to sending the data object identification request in the first message, a second message that includes a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the second message is addressed to the first user and the data object is not received in the second message
206

FIG. 2A

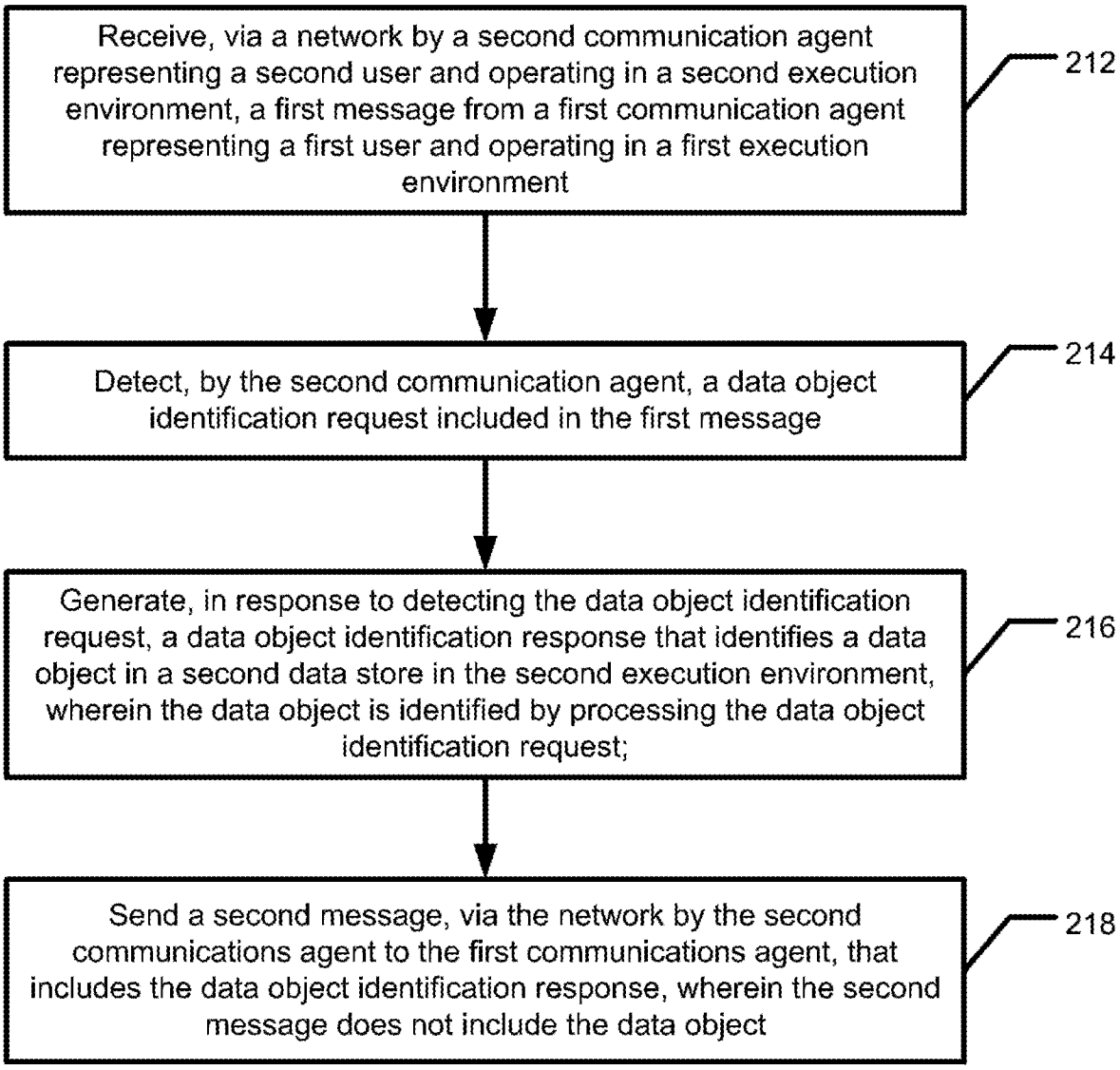

Receive, via a network by a second communication agent representing a second user and operating in a second execution environment, a first message from a first communication agent representing a first user and operating in a first execution environment
212
Detect, by the second communication agent, a data object identification request included in the first message
214
Generate, in response to detecting the data object identification request, a data object identification response that identifies a data object in a second data store in the second execution environment, wherein the data object is identified by processing the data object identification request;
216
Send a second message, via the network by the second communications agent to the first communications agent, that includes the data object identification response, wherein the second message does not include the data object
218

FIG. 2B

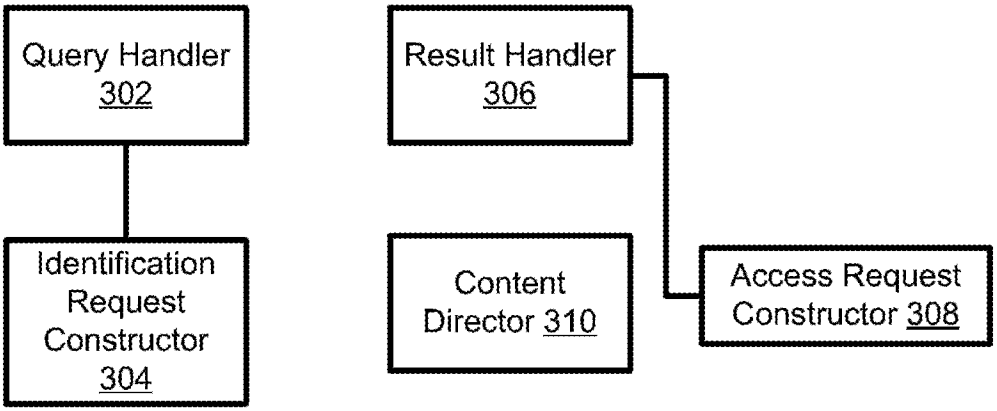

Query Handler 302
Result Handler 306
Identification Request Constructor 304
Content Director 310
Access Request Constructor 308

FIG. 3A

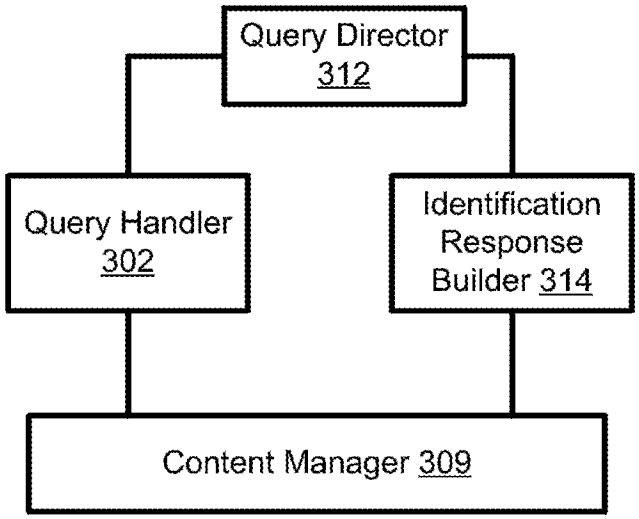

Query Director 312
Query Handler 302
Identification Response Builder 314
Content Manager 309

FIG. 3B

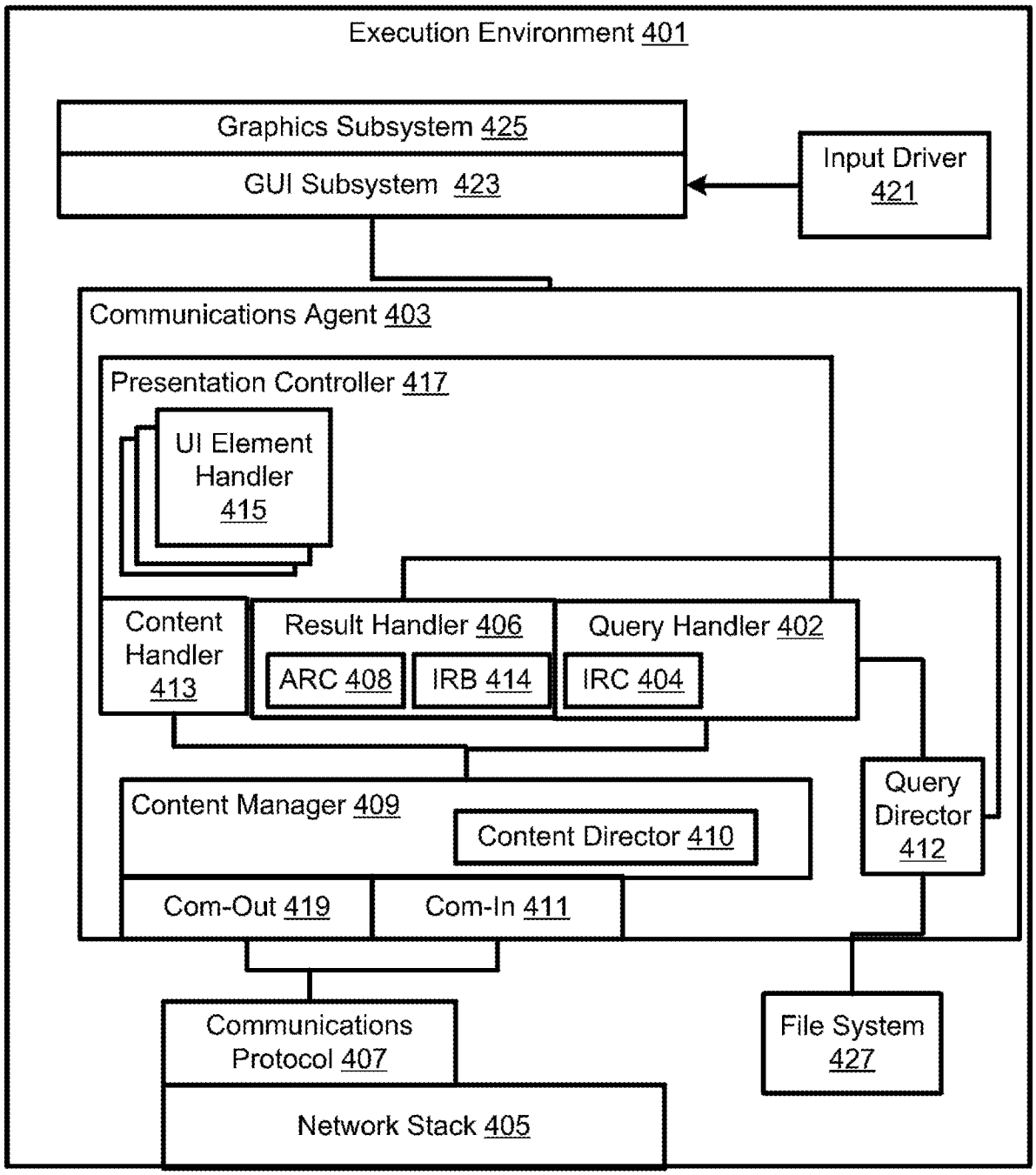

Execution Environment 401
Graphics Subsystem 425
GUI Subsystem 423
Input Driver 421
Communications Agent 403
Presentation Controller 417
UI Element Handler 415
Content Handler 413
Result Handler 406
ARC 408
IRB 414
Query Handler 402
IRC 404
Content Manager 409
Content Director 410
Query Director 412
Com-Out 419
Com-In 411
Communications Protocol 407
Network Stack 405
File System 427

FIG. 4

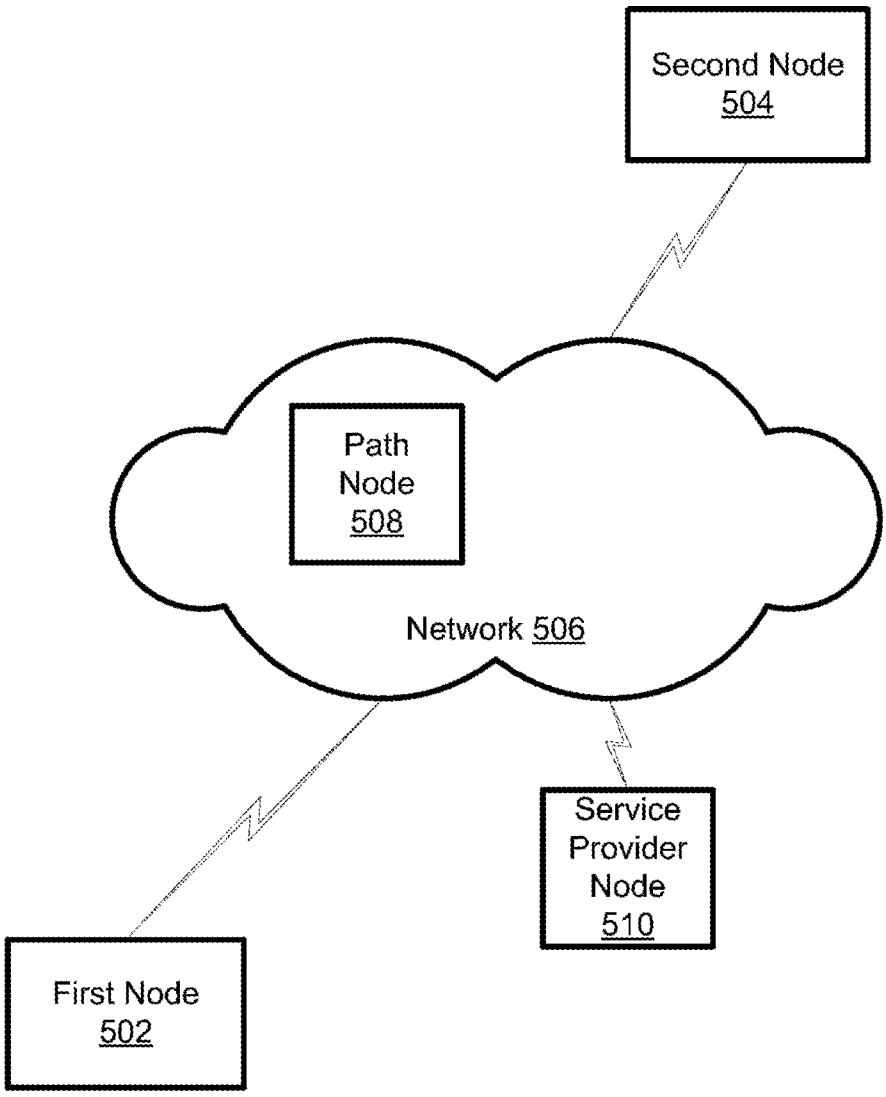

Second Node
504
Path
Node
508
Network 506
Service
Provider
Node
510
First Node
502

FIG. 5

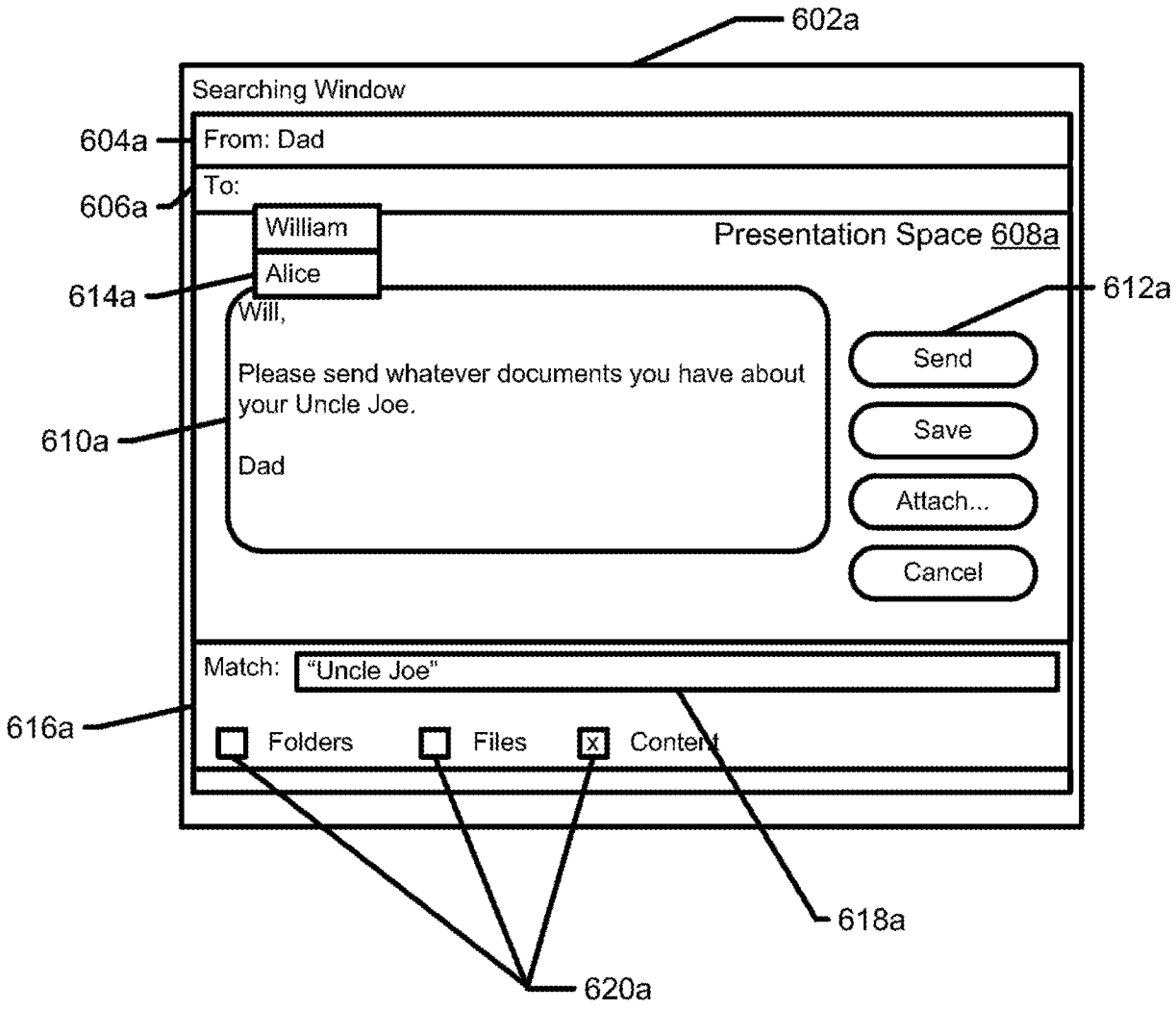

602a
Searching Window
604a
From: Dad
606a
To:
William
Alice
614a
Presentation Space 608a
612a
Will,
Please send whatever documents you have about your Uncle Joe.
610a
Dad
Send
Save
Attach...
Cancel
Match:
"Uncle Joe"
616a
Folders
Files
Content
618a
620a

FIG. 6A

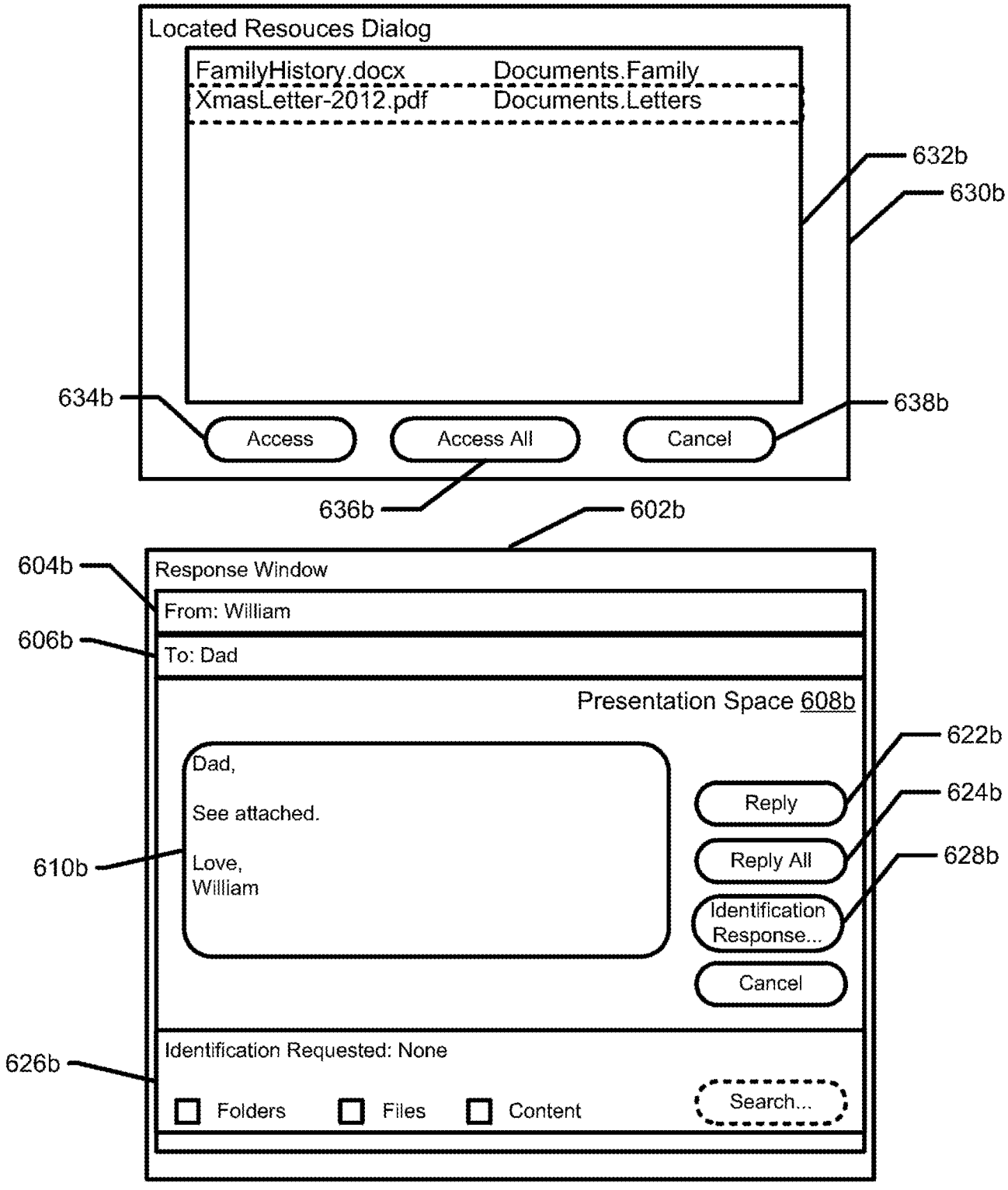

Located Resouces Dialog
FamilyHistory.docx Documents.Family
XmasLetter-2012.pdf Documents.Letters
632b
630b
634b
638b
Access
Access All
Cancel
636b
602b
604b
Response Window
From: William
606b
To: Dad
Presentation Space 608b
622b
624b
628b
610b
Dad,
See attached.
Love,
William
Reply
Reply All
Identification Response...
Cancel
626b
Identification Requested: None
Folders
Files
Content
Search...

FIG. 6B

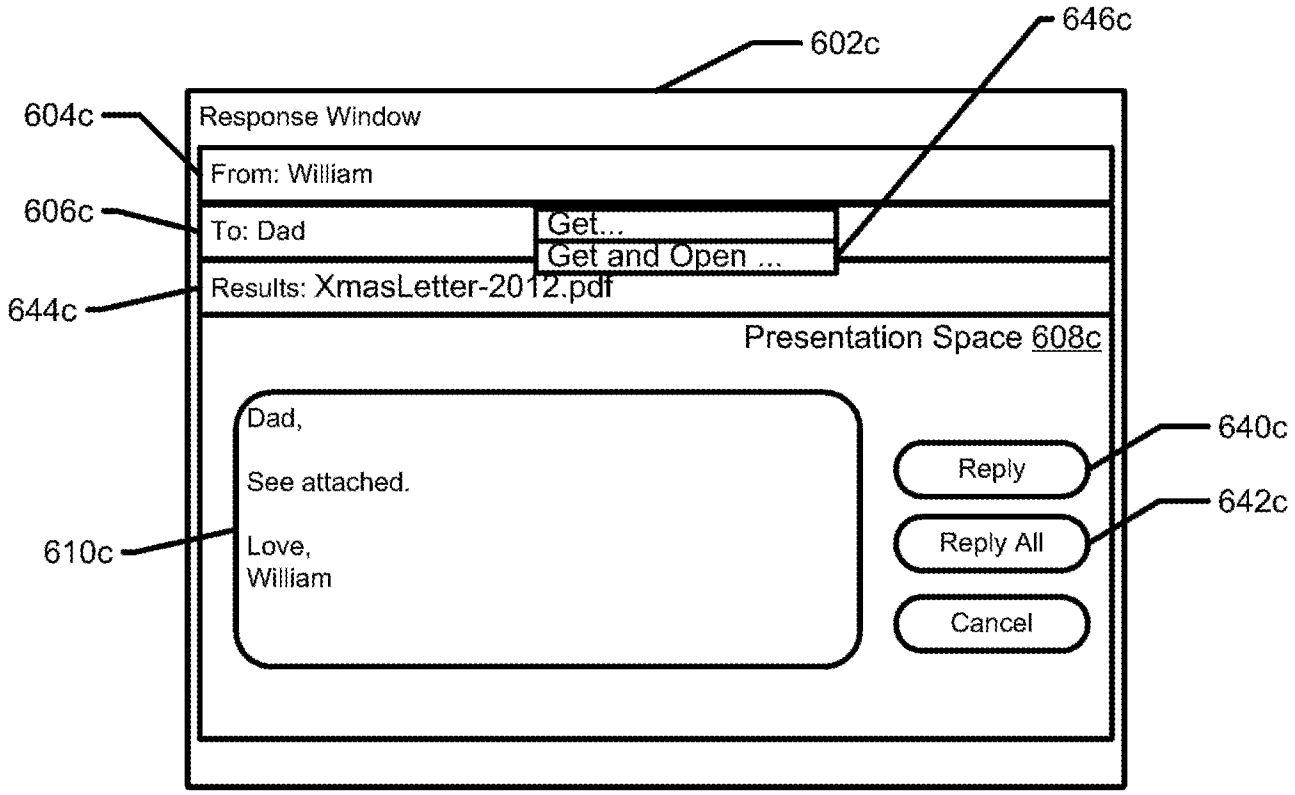

602c
646c
604c
Response Window
From: William
606c
To: Dad
Get...
Get and Open ...
644c
Results: XmasLetter-2012.pdf
Presentation Space 608c
Dad,
See attached.
Love,
William
610c
Reply
640c
Reply All
642c
Cancel

FIG. 6C

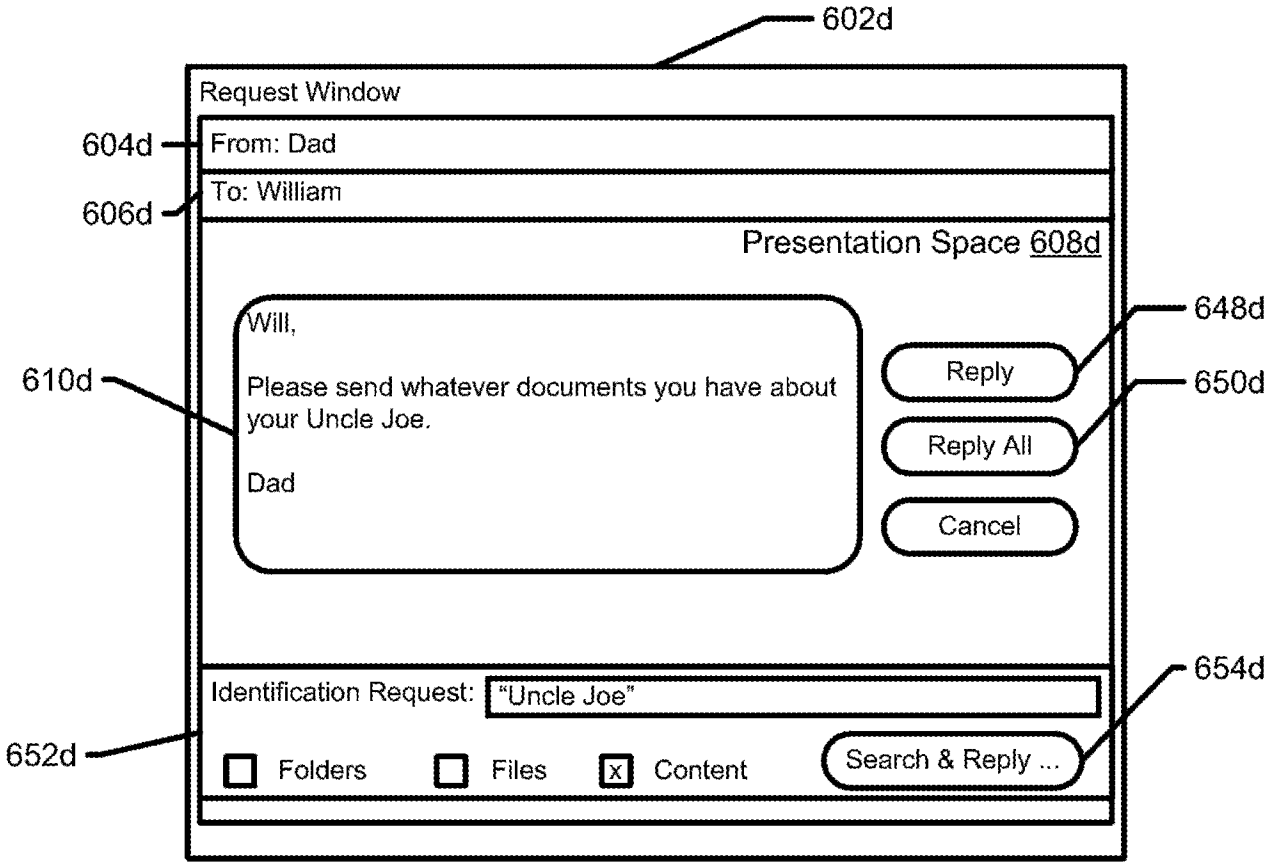

602d
Request Window
604d
From: Dad
606d
To: William
Presentation Space 608d
610d
Will,
Please send whatever documents you have about your Uncle Joe.
Dad
648d
Reply
650d
Reply All
Cancel
654d
Identification Request:
"Uncle Joe"
652d
Folders
Files
x
Content
Search & Reply ...

FIG. 6D

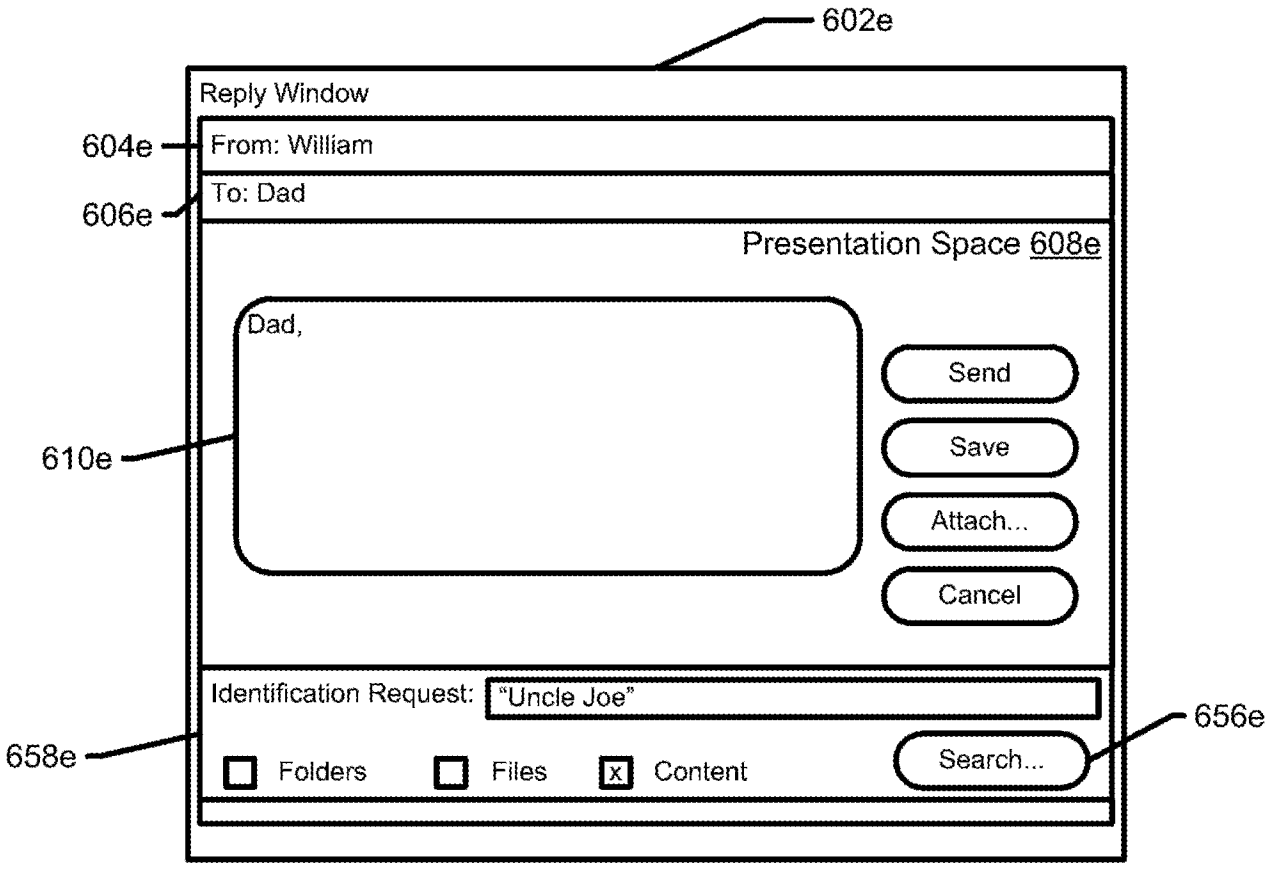

602e
Reply Window
604e
From: William
606e
To: Dad
Presentation Space 608e
Dad,
610e
Send
Save
Attach...
Cancel
Identification Request:
"Uncle Joe"
658e
656e
Folders
Files
x
Content
Search...

FIG. 6E

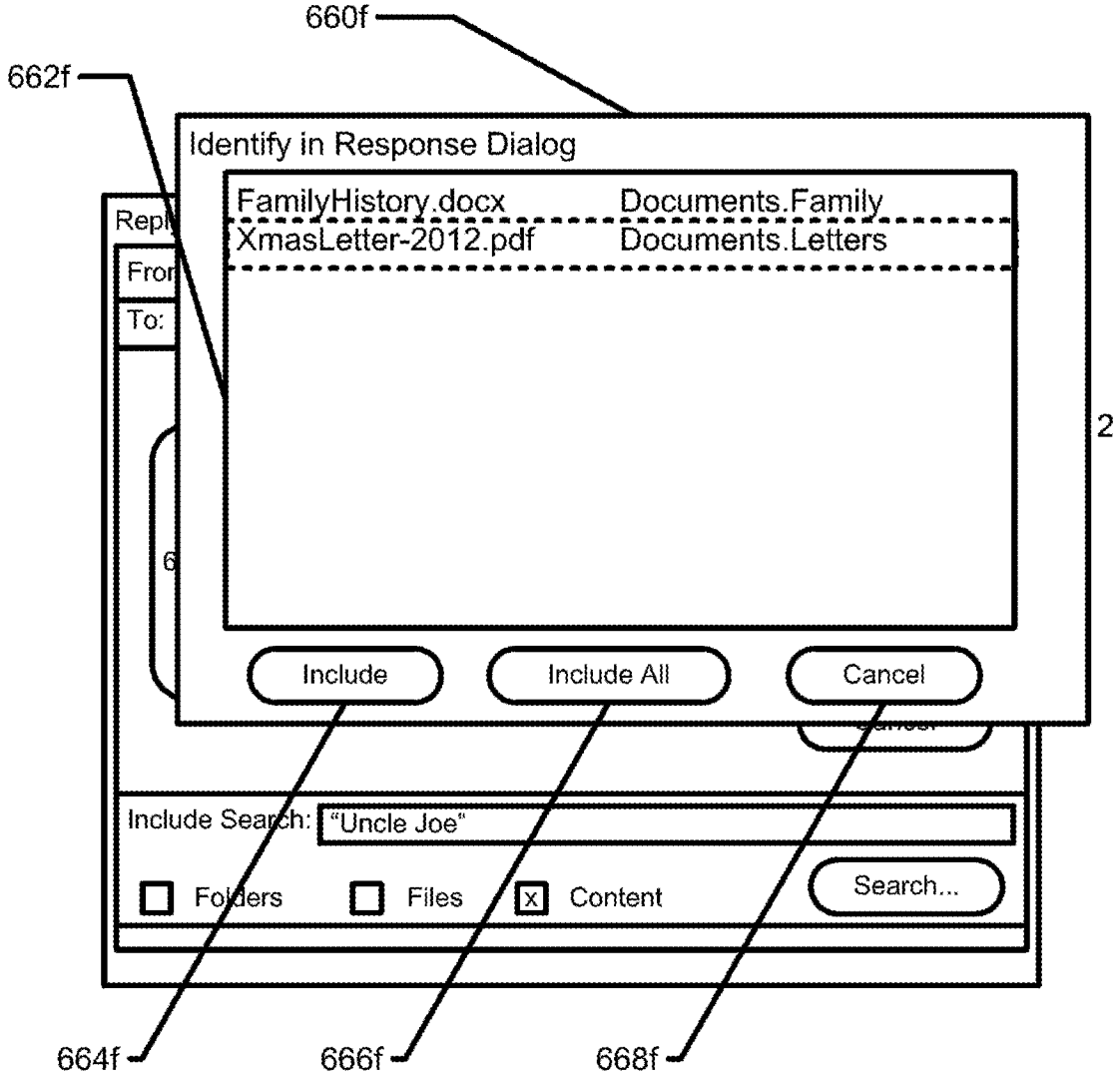

660f
662f
Identify in Response Dialog
FamilyHistory.docx Documents.Family
XmasLetter-2012.pdf Documents.Letters
Include
Include All
Cancel
Include Search: "Uncle Joe"
Folders
Files
Content
Search...
664f
666f
668f

FIG.6F

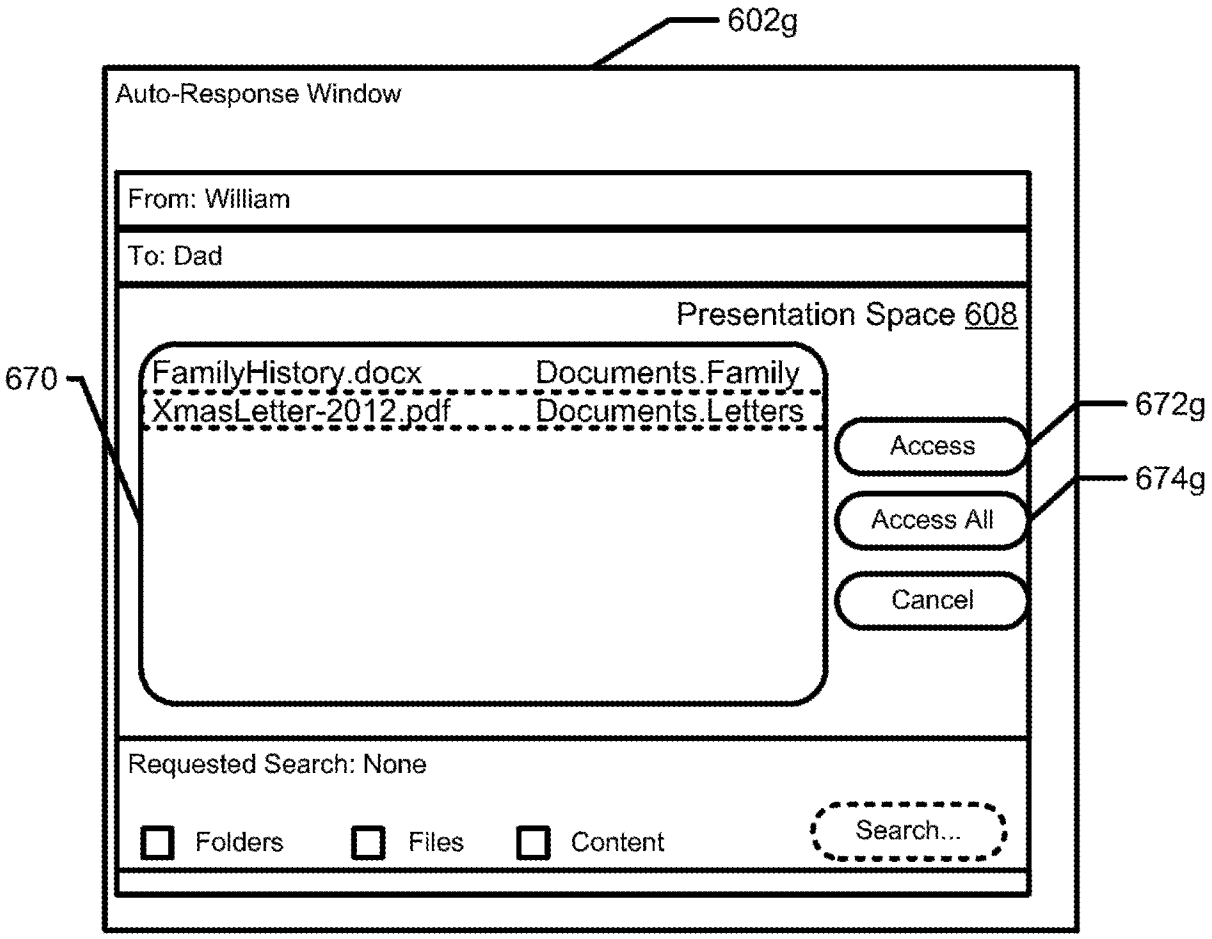

602g
Auto-Response Window
From: William
To: Dad
Presentation Space 608
670
FamilyHistory.docx Documents.Family
XmasLetter-2012.pdf Documents.Letters
Access
Access All
Cancel
672g
674g
Requested Search: None
Folders
Files
Content
Search...

FIG. 6G

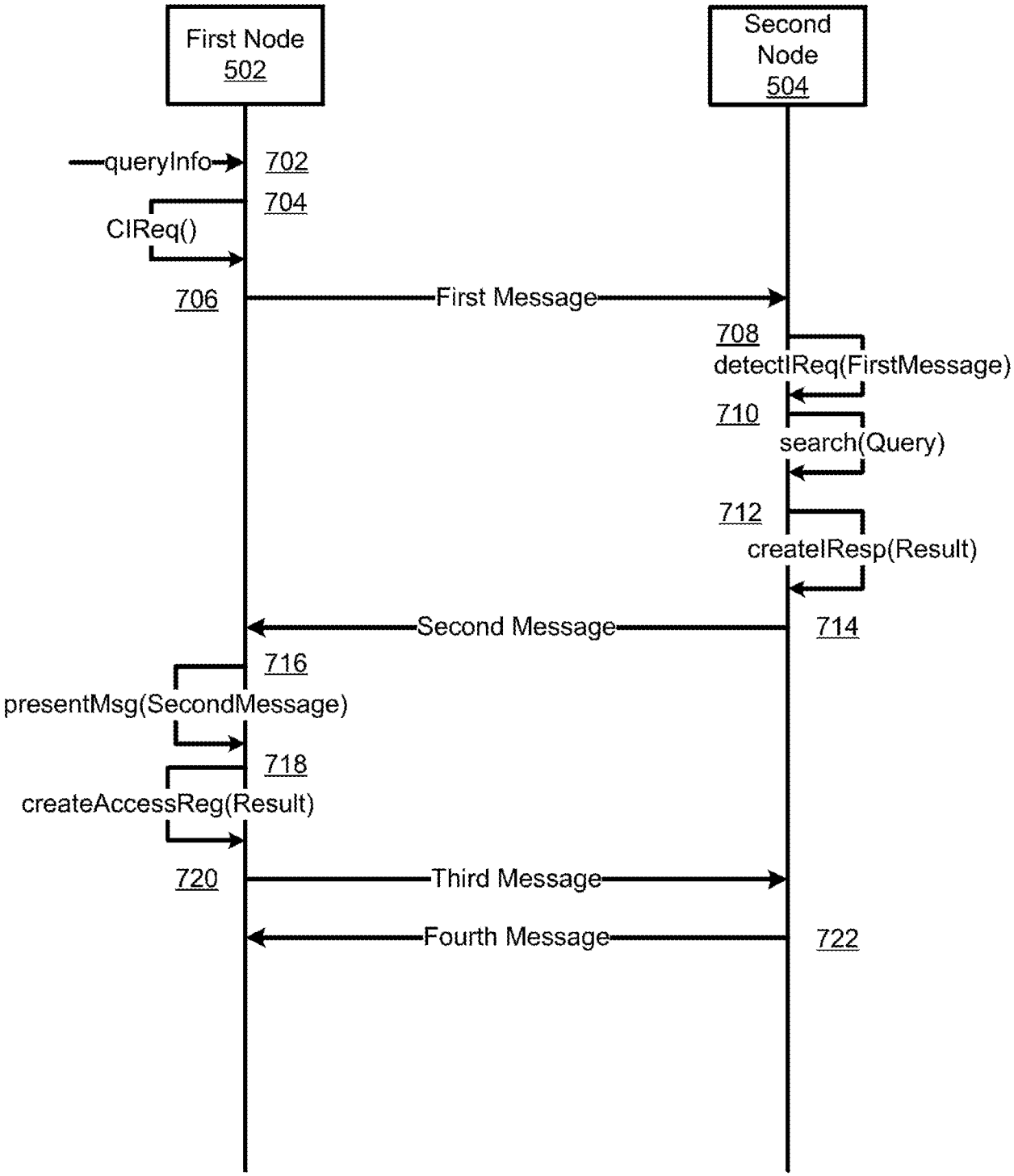

First Node
502
Second
Node
504
queryInfo
702
704
CIReq()
706
First Message
708
detectIReq(FirstMessage)
710
search(Query)
712
createIResp(Result)
Second Message
714
716
presentMsg(SecondMessage)
718
createAccessReg(Result)
720
Third Message
Fourth Message
722

FIG. 7

800a

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="--OoOT----toOo--------"

--OoOT----toOo--------
Content-Type: text/plain
Hi, this is the first message
--OoOT----toOo--------
Content-Type: application/id-request
Accept:  image/*
<id-request>
  <and>
    <criterion attribute="query" type="keyword" match-expression="'Uncle Joe'" />
    <criterion attribute="scope" resource-type="content" />
  </and>
</id-request>
--OoOT----toOo--------
```

814a, 812a, 804a, 806a, 810a, 808a, 802a

Fig. 8A

800b

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="--toOo----OoOt--------"

--toOo----OoOt---------
Content-Type: text/plain
Dad,

See attached.

Love,
William
--toOo----OoOt---------
Content-Type: application/query-respponse

FILE://Documents/Family/FamilyHistory.docx
FILE://Documents/Letters/XmasLetter-2012.pdf
--toOo----OoOt---------
```

802b, 806b, 804b

FIG. 8B