

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization

International Bureau



(43) International Publication Date
30 May 2013 (30.05.2013)

WIPO | PCT

(10) International Publication Number

WO 2013/078082 A1

(51) International Patent Classification:
G06F 15/16 (2006.01) *G06F 21/20* (2006.01)

(21) International Application Number:
PCT/US2012/065468

(22) International Filing Date:
16 November 2012 (16.11.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/301,039 21 November 2011 (21.11.2011) US

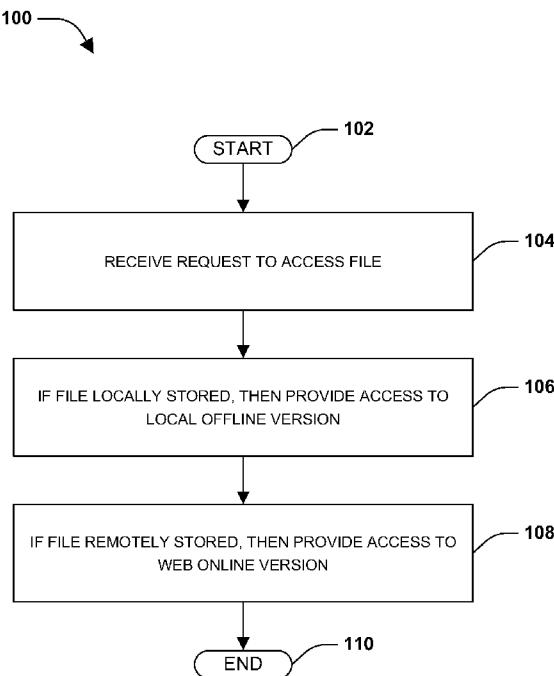
(71) Applicant: **MICROSOFT CORPORATION** [US/US];
One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **GARCIA-ASCANIO, Francisco**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).
BOEHM, Joshua Adam Alpern; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way,

Redmond, Washington 98052-6399 (US). **DIFFENBAUGH, David**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **FIORDALIS, Daniel Vincent**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **WIGGS, Joseph**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **IRWIN, Alyssa**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **SIGNER, Denise Elisabeth**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **LINKON, Adam**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **O'MAHONY, Allison**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **AKMAL, Mona**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **LUEDER, Rot**

[Continued on next page]

(54) Title: CLIENT APPLICATION FILE ACCESS



(57) **Abstract:** One or more techniques and/or systems are disclosed for providing a client application with access to a file based upon a local offline version of the file and/or web online version of the file. That is, a file may be shared/synchronized from a remote source, such as a web storage service. A web online version of the file may be maintained by the web storage service, while local offline versions of the file may be stored at client devices. A sync module comprising knowledge of the local and/or web versions of the file may provide a client application with access to the file based upon the local and/or web version of the file. The sync module may also delegate synchronization authorization to the client application given that the client application may comprise sophisticated file handling functionality.

FIG. 1



ger Allen; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **BOSE, Miko Arnab Sakhya Singha**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **BALASUBRAMANYAN, Balaji**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH,

GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

Published:

- with international search report (Art. 21(3))
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

CLIENT APPLICATION FILE ACCESS

BACKGROUND

[0001] Many users own a variety of computing devices, such as laptops, desktops, smart phones, and/or other devices. It may be desirable to share data between such devices. In one example, a web storage service may allow a user to upload files to online storage, and synchronize such files between various devices. In particular, a synchronization service, such as a sync module, may be installed on a user device. When files to be shared are available (e.g., not presently being accessed by a client application), the synchronization service may upload the shared files to the web storage service. Unfortunately, the synchronization service may comprise inefficient and/or inadequate file handling capabilities, at least as compared to a client application. For example, the synchronization service may comprise generic upload/download functionality that may not be tailored for particular types of files (e.g., whereas a word processing client application may be able to save changes to a text document in a desired, efficient, etc. manner, for example). Nevertheless, while a client application may comprise sophisticated file handling functionality, the client application (unlike the synchronization service) may, however, be unaware of the existence of the synchronization service, synchronization techniques that may be in place and/or the availability of offline access.

SUMMARY

[0002] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key factors or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

[0003] Among other things, one or more systems and/or techniques for providing a client application with access to a file, delegating synchronization authorization to the client application, and/or providing the client application with offline access to the file are disclosed herein. A client device hosting a client application may be associated with a sync module configured to access a web storage service hosting online storage. The online storage may be configured to store web online versions of shared files. The sync module may be configured to receive notifications from the web storage service regarding the web online versions of shared files (e.g., a notification that a web online version of a file was added, removed, and/or modified). The sync module may be configured to manage local offline versions of shared files (e.g., the sync module may add, remove,

and/or modify local copies of files shared through the web storage service). In this way, the sync module may comprise knowledge, such as synchronization information, associated with local offline versions of files and/or web online versions of files (e.g., where client applications do not possess such knowledge). Accordingly, the sync module

5 may provide synchronization information to a client application (e.g., a URL of a web online version of a file, a local file path of a local offline version of the file, synchronization authorization, etc.), such that the client application may perform synchronization of files, access a file based upon a local offline version of the file and/or a web online version of the file, and/or access shared files even if the web storage service
10 and/or the client device may be offline (e.g., offline access may be provided to previously synchronized files). It may be appreciated that synchronization of files performed by the client application may range from merely transferring file data (e.g., transferring files up to and/or down from online storage) to performing sophisticated synchronization functionality (e.g., synchronizing content between files). That is, synchronization and/or
15 the like may merely comprise transferring data as opposed to synchronizing data files, although it could, of course, comprise synchronizing as well.

[0004] In one example of providing a client application with access to a file, a request from a client application to access a file may be received. For example, a word processor may generate a request comprising a local file path associated with a local offline version 20 of a shared text file, a URL associated with a web online version of the shared text file, and/or a general inquiry requesting an enumeration of shared text files. If a local offline version of the file is stored locally, then the client application may be provided with access to the local offline version as the file (e.g., the word processor may be provided with access to a text file derived at least in part from a local offline version of the text file). In
25 one example, access may be provided to the local offline version of the file even if a web online version of the file is unavailable (e.g., offline). If a web online version of the file is stored remotely, then the client application may be provided with access to the web online version of the file as the file (e.g., the word processor may be provided with access to the text file derived at least in part from a web online version of the text file).

30 **[0005]** In one example, the client application may be provided with access to both the local offline version of the file and the web online version of the file as the file (e.g., the word processor may be provided with access to the text file derived from at least some of the local offline version of the text file and at least some of the web online version of the text file, such that the text file accessed by the word processor may be associated with both

the local offline version of the text file and the web online version of the text file). In this way, interactions with and/or modifications to the file by the client application may be synchronized between the file accessed by the client application, the local offline version of the file, and/or the web online version of the file. For example, synchronization may be

5 performed real-time while the client application is accessing the file. Such real-time synchronization may enable co-authoring between the file and a corresponding file accessed by a remote client application. For example, the file and/or the corresponding file may be updated in real-time based upon the web online version of the file accessible to the client application and/or the remote client application.

10 [0006] In one example of delegating synchronization authorization to a client application, a client application configured to interact with a particular file type with which a file is formatted may be identified (e.g., a word processor may be configured to interact with text files formatted as a .txt file type). The client application may comprise sophisticated file handling functionality with regard to the file type. For example, a word processor may be configured to merely save changes made to an existing text file, as opposed to resaving the entire text file. Because a plethora of different file types may be shared by the web storage service (e.g., .txt text files, .bmp image files, .zip compressed files, .xls spreadsheet files, etc.), the sync module may not comprise adequate file handling functionality with regard to the various file types (e.g., the sync module may merely

15 comprise generic download/upload functionality). Accordingly, it may be efficient for the sync module to delegate synchronization authorization to the client application so that the client application may synchronize a file accessed by the client application, a local offline version of the file, and/or a web online version of the file. It may be appreciated that aspects herein generally promote, leverage, etc. strengths, advantages, etc. associated with

20 synchronization services and client applications, for example, while mitigating, circumventing, etc. weaknesses, disadvantages, etc. associated with synchronization services and client applications, for example. It may be appreciated that in one example, delegating synchronization authorization may comprise merely delegating the task of synchronization (e.g., file transfer) to the client application (e.g., as opposed to performing

25 synchronization services and client applications, for example, while mitigating, circumventing, etc. weaknesses, disadvantages, etc. associated with synchronization services and client applications, for example. It may be appreciated that in one example, delegating synchronization authorization may comprise merely delegating the task of synchronization (e.g., file transfer) to the client application (e.g., as opposed to performing

30 extensive security functionality to grant authorization access to files). That is, (unless indicated to the contrary) even though delegating synchronization ‘authorization’ and/or the like may be used herein throughout, the instant application, including the scope of the appended claims, is not meant to be limited to and/or require authorizing actions (e.g., requesting and/or verifying credentials, etc.).

[0007] To the accomplishment of the foregoing and related ends, the following description and annexed drawings set forth certain illustrative aspects and implementations. These are indicative of but a few of the various ways in which one or more aspects may be employed. Other aspects, advantages, and novel features of the disclosure will become apparent from the following detailed description when considered in conjunction with the annexed drawings.

DESCRIPTION OF THE DRAWINGS

[0008] Fig. 1 is a flow chart illustrating an exemplary method of providing a client application with access to a file.

10 [0009] Fig. 2 is a flow chart illustrating an exemplary method of delegating synchronization authorization to a client application.

[0010] Fig. 3 is a flow chart illustrating an exemplary method of providing a client application with offline access to a file.

15 [0011] Fig. 4 is a component block diagram illustrating an exemplary system for delegating synchronization authorization to a client application.

[0012] Fig. 5 is a component block diagram illustrating an exemplary system for providing a client application with access to a file.

[0013] Fig. 6 is a component block diagram illustrating an exemplary system for providing a client application with offline access to a file.

20 [0014] Fig. 7 is an illustration of an example of enabling co-authoring of a document.

[0015] Fig. 8 is an illustration of an exemplary computer-readable medium wherein processor-executable instructions configured to embody one or more of the provisions set forth herein may be comprised.

25 [0016] Fig. 9 illustrates an exemplary computing environment wherein one or more of the provisions set forth herein may be implemented.

DETAILED DESCRIPTION

[0017] The claimed subject matter is now described with reference to the drawings, wherein like reference numerals are generally used to refer to like elements throughout. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the claimed subject matter. It may be evident, however, that the claimed subject matter may be practiced without these specific details. In other instances, structures and devices are illustrated in block diagram form in order to facilitate describing subject matter.

[0018] Today, data is commonly shared between computing devices. For example, a user may share photos between a desktop device and a smart phone using online storage hosted by a web storage service. Synchronization of the shared photos may be provided by a sync module on the desktop device and the smart phone. Unfortunately, if the web storage service is unavailable (e.g., offline), then the user may be unable to access the shared photos stored within the online storage from the desktop device and/or the smart phone. Additionally, the sync module may be unfamiliar with the file type associated with the shared photos (e.g., the sync module may merely comprise generic upload/download functionality), whereas a photo client application on the desktop device and/or smart

5 phone may be better suited for handling storage/synchronization functionality associated with the shared photos. However, the photo client application may be unaware of the web storage service and/or the synchronization policy that may be in place. Thus, without leveraging the sophisticated file handling functionality of the photo client application, the sync module may be inefficient at synchronizing the shared photos.

10 **[0019]** Accordingly, among other things, one or more systems and/or techniques for providing a client application with access to a file, delegating synchronization authorization to the client application, and/or providing the client application with offline access to the file are provided herein.

[0020] One embodiment of providing a client application with access to a file is illustrated by an exemplary method 100 in Fig. 1. At 102, the method starts. At 104, a request from a client application to access a file may be received. In one example of receiving the request, the request may comprise an inquiry to a sync module about available files shared by a web storage service. The sync module may determine one or more file types that the client application may be able to understand and/or interact with (e.g., the client application may provide file type compatibility information to the sync module). In this way, the client application may be provided with access to one or more files formatted according to file types corresponding to the identified file types. In another example of receiving the request, the request may comprise a local file path associated with the file shared by the web storage service. In another example, the request may comprise a URL associated with the file shared by the web storage service. In another example, the request may comprise a resourceId identifying the file (e.g., the resourceId may uniquely identify the file with the sync module, the web storage service, and/or the client application). It may be appreciated that in one example, the sync module may

comprise knowledge regarding files shared by the web storage service and/or locally stored versions of shared files, whereas the client application may lack such knowledge.

[0021] At 106, if a local offline version of the file is stored locally, then the client application may be provided with access to the local offline version as the file. For example, a local file path associated with the local offline version of the file may be provided to the client application (e.g., the sync module may derive the local file path based upon a URL associated with the web online version that may have been specified within the request by the client application). In this way, the client application may open the file from the local offline version using the local file path. In one example, if the file is designated to be stored locally, but is not yet stored locally, then the sync component may provide the client application with a local path at which a new local offline version of the file is to be created by the client application. At 108, if a web online version of the file is stored remotely (e.g., within online storage hosted by the web storage service), then the client application may be provided with access to the web online version of the file as the file. For example, a URL associated with the web online version of the file may be provided to the client application (e.g., the sync module may derive the URL based upon a local file path associated with the local offline version that may have been specified within the request by the client application). In this way, the client application may open the file from the web online version using the URL. In one example, if the file is designated to be stored remotely, but is not yet stored remotely, then the sync component may provide the client application with a remote path at which a new remote version of the file is to be created by the client application.

[0022] In one example, the client application may be provided with merely access to the local offline version as the file because the web online version of the file may not be available (e.g., offline) and/or may be inefficient for the client application to access. In another example, the client application may be provided with merely access to the web online version as the file because the web online version may be an up-to-date version of the file and/or a local offline version of the file may be unavailable. In another example, the client application may be provided with access to both the local offline version of the file and the web online version of the file as the file. That is, the file accessed by the client application may be derived from and/or synchronously associated with at least some of the local offline version of the file and at least some of the web online version of the file. In this way, the file accessed by the client application, the local offline version of the file, and/or the web online version of the file may be synchronized. Such synchronization may,

for example, enable co-authoring of the file between the client application and a remote client application accessing a corresponding file. For example, the file and the corresponding file may be synchronized through the web online version of the file (e.g., modifications made to the web online version of the file by the remote client application

5 may be detected by the sync module, and the sync module may instruct the client application to synchronize the file with the web online version of the file).

[0023] The client application may comprise sophisticated functionality for handling certain file types, whereas the sync module may not comprise such advanced functionality (e.g., the sync module may merely comprise generic upload/download functionality due to

10 the wide variety of file types that may be associated with shared files). Accordingly,

synchronization authorization may be delegated to the client application. In one example,

the client application may be provided with a local file path associated with the local

offline version of the file, and may be delegated synchronization authorization to

15 synchronize the file accessed by the client application with the local offline version of the

file using the local file path. In another example, the client application may be provided

with a URL associated with the web online version of the file, and may be delegated

synchronization authorization to synchronize the file accessed by the client application

with the web online version of the file using the URL. In another example, the client

application may be delegated synchronization authorization to synchronize the file

20 accessed by the client, the local offline version of the file, and/or the web online version of the file. In this way, file handling functionality provided by the client application may be leveraged to provide enhanced synchronization.

[0024] Because the sync module may subscribe to notifications from the web storage service, the sync module may provide a notification to the client application that the web

25 online version of the file was modified, and may instruct the client application to

synchronize the file accessed by the client application and the web online version of the

file. Because the sync module may monitor local storage of shared local files, the sync

module may instruct the client application to synchronize the file accessed by the client

application and the local offline version of the file. At 110, the method ends.

30 [0025] One embodiment of delegating synchronization authorization to a client application is illustrated by an exemplary method 200 in Fig. 2. At 202, the method starts. At 204, a client application configured to interact with a particular file type with which a file is formatted may be identified. For example, an operating system registry may be consulted to determine whether the client application can open files formatted according to

the file type (e.g., a registry key may indicate that a word processor may be capable of opening a .txt file). In another example, an installation of the client application, an un-installation of the client application, a modification to the client application and/or a file type (e.g., .txt and .docx (e.g., but not .jpg)) associated with the client application may be

5 identified using one or more registry keys within the operating system registry. At 206, synchronization authorization may be delegated to the client application to synchronize the file accessed by the client application and a local offline version of the file and/or a web online version of the file. In one example, the file may be identified by a resourceId (e.g., the resourceId may uniquely identify the file with the sync module, the web storage

10 service, and/or the client application).

[0026] In one example, the client application may be provided with a local file path associated with the local offline version of the file, and may be delegated synchronization authorization to synchronize the file accessed by the client application with the local offline version of the file using the local file path. In another example, the client

15 application may be provided with a URL associated with the web online version of the file, and may be delegated synchronization authorization to synchronize the file accessed by the client application with the web online version of the file using the URL. In another example, the client application may be delegated synchronization authorization to synchronize the file accessed by the client, the local offline version of the file, and/or the

20 web online version of the file. In this way, file handling functionality provided by the client application may be leveraged to provide enhanced synchronization. At 208, the method ends.

[0027] One embodiment of providing a client application with offline access to a file is illustrated by an exemplary method 300 in Fig. 3. At 302, the method starts. At 304, a

25 request to access a file may be received from a client application. In one example, the request may comprise a resourceId identifying the file (e.g., the resourceId may uniquely identify the file with the sync module, the web storage service, and/or the client application). Optionally, at 306, unavailability of a web online version of the file may be determined (e.g., a web storage service hosting online storage comprising the web online

30 version of the file may be inaccessible). It may be appreciated that in one example, a sync module may have previously stored a local offline version of the file (e.g., the sync module may have downloaded the web online version of the file as the local offline version of the file stored locally to the client application). At 308, the client application may be provided with access to the local offline version of the file. For example, a local

file path associated with the local offline version of the file may be provided to the client application (e.g., the request may comprise a URL associated with the web online version of the file, and the local file path may be determined based upon the URL (e.g., because the online version is not available)). At 310, the method ends.

5 [0028] Fig. 4 illustrates an example of a system 400 configured for delegating synchronization authorization to a client application 410. The system 400 may comprise a sync module 416. The sync module 416 may be configured to subscribe to notifications 408 from a remote source 402, such as a web storage service hosting online storage of shared files. The sync module 416 may receive notifications 408 comprising a variety of 10 information associated with shared/synchronized files, such as notifications that a web online version of a file 404 was created, modified, and/or deleted. The sync module 416 may be configured to monitor and/or perform topology events 420 (e.g., create, modify, delete, etc.) associated with local storage 422 (e.g., storage associated with a computing device comprising client application 410) of shared/synchronized files, such as a local 15 offline version of a file 424. In one example, file 404 may be identified by a resourceId (e.g., the resourceId may uniquely identify the file 404 with the sync module 416, the web storage service, and/or the client application 410).

[0029] Because the sync module 416 may not comprise sophisticated functionality to handle particular types of files (e.g., functionality used to determine what data within a file 20 to synchronize), the sync module 406 may be configured to provide sync information 414 to the client application 410. The client application 410 may comprise sophisticated functionality for handling files formatted according to particular file types. In this way, the client application 410 may perform synchronization, such as synchronization 406 and/or 418, between a file 412 accessed by the client application 410, the web online 25 version of the file 404, and/or the local offline version of the file 424. The sync module 416 may provide synchronization information to the client application 410 (e.g., information regarding the existence of the sync module 416, where local offline versions of files are stored, etc.). In one example, the sync module 416 may provide a URL 30 associated with the web online version of the file 404 and synchronization authorization within the sync information 414, which may allow the client application 410 to synchronize 406 the file 412 and the web online version of the file 404, for example. In another example, the sync module 416 may provide a local file path associated with the local offline version of the file 424 and synchronization authorization within the sync information 414, which may allow the client application 410 to synchronize 418 the file

412 and the local offline version of the file 424. In another example, the sync module 416 may provide the local file path, the URL, and/or synchronization authorization within the sync information 414, which may allow the client application 410 to synchronize the file 412, the local offline version of the file 424, and/or the web online version of the file 404.

5 The sync module 416 may poll the client application 410 for information regarding synchronization progress, success, and/or failure (e.g., the client application 410 may notify the sync module 416 of new sync events that may be available for polling, which may provide for, among other things, an increase in sync module UI updates with little to no additional overhead, for example). In this way, the client application 410 may perform 10 robust synchronization on behalf of the sync module 416 (e.g., because the sync module 416 may not be equipped, configured, etc. to handle sophisticated synchronization).

[0030] Fig. 5 illustrates an example of a system 500 configured for providing a client application 508 with access to a file 514. The system 500 may comprise a sync module 510. The sync module 510 may be configured to subscribe to notifications from a remote 15 source 502, such as a web storage service hosting online storage of shared/synchronized files. The sync module 510 may receive information regarding the shared/synchronized files, such as a notification regarding a web online version of the file 504 (e.g., a notification comprising information regarding the creation, modification, and/or deletion of the web online version of the file 504). The sync module 510 may be configured to 20 monitor and/or perform topology events (e.g., create, modify, delete, etc.) associated with local storage 516 (e.g., storage associated with a computing device comprising client application 508) of shared/synchronized files, such as a local offline version of the file 518.

[0031] The sync module 510 may be configured to provide sync information 512, such 25 as a URL associated with the web online version of the file 504 and/or a local file path associated with the local offline version of the file 518, to the client application 508. In this way, the sync module 510 may provide the client application 508 with access 520 to the local offline version of the file 518 as the file 514 and/or with access 506 to the web online version of the file 504 as the file 514. The client application 508 may utilize the 30 sync information 512 when accessing the file 514. For example, the file 514 may be opened by the client application 508 utilizing at least some of the web online version of the file 504 and/or at least some of the local offline version of the file 518.

[0032] Fig. 6 illustrates an example of a system 600 configured for providing a client application 608 with offline access to a file 614. The system 600 may comprise a sync

module 610. The sync module 610 may be configured to subscribe to notifications from a remote source 602, such as a web storage service hosting online storage of shared/synchronized files. The sync module 610 may receive information regarding the shared/synchronized files, such as a notification regarding a web online version of the file

5 604 (e.g., a notification comprising information regarding the creation, modification, and/or deletion of the web online version of the file 604). The sync module 610 may be configured to locally store versions of shared/synchronized files within local storage 616 (e.g., storage locally accessible to the client application 608). For example, the sync module 610 may store the web online version of the file 604 within the local storage 616 as a local offline version of the file 618. In this way, the sync module 610 may provide the client application 608 with offline access to the file 614 based upon the local offline version of the file 618.

10

[0033] In one example, the sync module 610 may receive a request from the client application 608 to access to the file 614. The sync module 610 may provide the client application 608 with a local file path 612 associated with the local offline version of the file 618. Because the remote source 602 and/or the web online version of the file 604 may be offline 606 (e.g., unavailable to the client application 608), the sync module 610 may merely provide the local file path 612 to the client application 608. The client application 608 may thus access 620 the local offline version of the file 618 as the file 614 using the local file path 610. In this way, the client application 608 may be provided with offline access to the file 614.

20

[0034] Fig. 7 illustrates an example 700 of enabling co-authoring of a document. A document, such as a text file, may be uploaded to a remote source 702, such as a web storage service hosting online storage of shared/synchronized files, as a web online version of a file 704. The web online version of the file 704 may be shared/synchronized between a client application 710, a remote client application 712, and/or other client applications and/or devices (not shown). The web online version of the file 704 may be synchronized 706 to a client device hosting the client application 710 and/or may be synchronized 708 to a remote client device hosting the remote client application 712. The client application 710 may access the web online version of the file 704 as a file 714 (e.g., an instance of the web online version of the file 704). The remote client application 712 may access the web online version of the file 704 as a corresponding file 716 (e.g., an instance of the web online version of the file 704). Because the file 714 and the corresponding file 718 may be synchronized with the web online version of the file 704

25

30

while being accessed by the client application 710 and/or the remote client application 712, co-authoring may be enabled. For example, while a user of client application 710 interacts with file 714, the user may be able to see interactions made by a remote user of remote client application 712 with the corresponding file 716. Similarly, while the remote user of remote client application 712 interacts with the corresponding file 716, the remote user may be able to see interactions made by the user with the file 714. In this way, co-authoring may be enabled based upon synchronization 706 and/or synchronization 708 of the web online version of the file 704.

[0035] Still another embodiment involves a computer-readable medium comprising processor-executable instructions configured to implement one or more of the techniques presented herein. An exemplary computer-readable medium that may be devised in these ways is illustrated in Fig. 8, wherein the implementation 800 comprises a computer-readable medium 816 (e.g., a CD-R, DVD-R, or a platter of a hard disk drive), on which is encoded computer-readable data 814. This computer-readable data 814 in turn comprises a set of computer instructions 812 configured to operate according to one or more of the principles set forth herein. In one such embodiment 800, the processor-executable computer instructions 812 may be configured to perform a method 810, such as at least some of the exemplary method 100 of Fig. 1, at least some of exemplary method 200 of Fig. 2, and/or at least some of the exemplary method 300 of Fig. 3, for example. In

another such embodiment, the processor-executable instructions 812 may be configured to implement a system, such as at least some of the exemplary system 400 of Fig. 4, at least some of exemplary system 500 of Fig. 5, and/or at least some of exemplary system 600 of Fig. 6, for example. Many such computer-readable media may be devised by those of ordinary skill in the art that are configured to operate in accordance with the techniques presented herein.

[0036] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

[0037] As used in this application, the terms "component," "module," "system", "interface", and the like are generally intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component may be, but is not limited to being, a process running on a

processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a controller and the controller can be a component. One or more components may reside within a process and/or thread of execution and a component may be localized on one computer and/or

5 distributed between two or more computers.

[0038] Furthermore, the claimed subject matter may be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof to control a computer to implement the disclosed subject matter. The term "article of manufacture" 10 as used herein is intended to encompass a computer program accessible from any computer-readable device, carrier, or media. Of course, those skilled in the art will recognize many modifications may be made to this configuration without departing from the scope or spirit of the claimed subject matter.

[0039] Fig. 9 and the following discussion provide a brief, general description of a 15 suitable computing environment to implement embodiments of one or more of the provisions set forth herein. The operating environment of Fig. 9 is only one example of a suitable operating environment and is not intended to suggest any limitation as to the scope of use or functionality of the operating environment. Example computing devices include, but are not limited to, personal computers, server computers, hand-held or laptop 20 devices, mobile devices (such as mobile phones, Personal Digital Assistants (PDAs), media players, and the like), multiprocessor systems, consumer electronics, mini computers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

[0040] Although not required, embodiments are described in the general context of 25 "computer readable instructions" being executed by one or more computing devices. Computer readable instructions may be distributed via computer readable media (discussed below). Computer readable instructions may be implemented as program modules, such as functions, objects, Application Programming Interfaces (APIs), data structures, and the like, that perform particular tasks or implement particular abstract data 30 types. Typically, the functionality of the computer readable instructions may be combined or distributed as desired in various environments.

[0041] Fig. 9 illustrates an example of a system 910 comprising a computing device 912 configured to implement one or more embodiments provided herein. In one configuration, computing device 912 includes at least one processing unit 916 and

memory 918. Depending on the exact configuration and type of computing device, memory 918 may be volatile (such as RAM, for example), non-volatile (such as ROM, flash memory, etc., for example) or some combination of the two. This configuration is illustrated in Fig. 9 by dashed line 914.

5 [0042] In other embodiments, device 912 may include additional features and/or functionality. For example, device 912 may also include additional storage (e.g., removable and/or non-removable) including, but not limited to, magnetic storage, optical storage, and the like. Such additional storage is illustrated in Fig. 9 by storage 920. In one embodiment, computer readable instructions to implement one or more embodiments provided herein may be in storage 920. Storage 920 may also store other computer readable instructions to implement an operating system, an application program, and the like. Computer readable instructions may be loaded in memory 918 for execution by processing unit 916, for example.

10

15 [0043] The term “computer readable media” as used herein includes computer storage media. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions or other data. Memory 918 and storage 920 are examples of computer storage media. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, Digital 20 Versatile Disks (DVDs) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by device 912. Any such computer storage media may be part of device 912.

25 [0044] Device 912 may also include communication connection(s) 926 that allows device 912 to communicate with other devices. Communication connection(s) 926 may include, but is not limited to, a modem, a Network Interface Card (NIC), an integrated network interface, a radio frequency transmitter/receiver, an infrared port, a USB connection, or other interfaces for connecting computing device 912 to other computing devices. Communication connection(s) 926 may include a wired connection or a wireless 30 connection. Communication connection(s) 926 may transmit and/or receive communication media.

[0045] The term “computer readable media” may include communication media. Communication media typically embodies computer readable instructions or other data in a “modulated data signal” such as a carrier wave or other transport mechanism and

includes any information delivery media. The term “modulated data signal” may include a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal.

[0046] Device 912 may include input device(s) 924 such as keyboard, mouse, pen, 5 voice input device, touch input device, infrared cameras, video input devices, and/or any other input device. Output device(s) 922 such as one or more displays, speakers, printers, and/or any other output device may also be included in device 912. Input device(s) 924 and output device(s) 922 may be connected to device 912 via a wired connection, wireless connection, or any combination thereof. In one embodiment, an input device or an output 10 device from another computing device may be used as input device(s) 924 or output device(s) 922 for computing device 912.

[0047] Components of computing device 912 may be connected by various interconnects, such as a bus. Such interconnects may include a Peripheral Component Interconnect (PCI), such as PCI Express, a Universal Serial Bus (USB), firewire (IEEE 15 1394), an optical bus structure, and the like. In another embodiment, components of computing device 912 may be interconnected by a network. For example, memory 918 may be comprised of multiple physical memory units located in different physical locations interconnected by a network.

[0048] Those skilled in the art will realize that storage devices utilized to store 20 computer readable instructions may be distributed across a network. For example, a computing device 930 accessible via a network 928 may store computer readable instructions to implement one or more embodiments provided herein. Computing device 912 may access computing device 930 and download a part or all of the computer readable instructions for execution. Alternatively, computing device 912 may download pieces of 25 the computer readable instructions, as needed, or some instructions may be executed at computing device 912 and some at computing device 930.

[0049] Various operations of embodiments are provided herein. In one embodiment, one or more of the operations described may constitute computer readable instructions stored on one or more computer readable media, which if executed by a computing device, 30 will cause the computing device to perform the operations described. The order in which some or all of the operations are described should not be construed as to imply that these operations are necessarily order dependent. Alternative ordering will be appreciated by one skilled in the art having the benefit of this description. Further, it will be understood that not all operations are necessarily present in each embodiment provided herein.

[0050] Moreover, the word "exemplary" is used herein to mean serving as an example, instance, or illustration. Any aspect or design described herein as "exemplary" is not necessarily to be construed as advantageous over other aspects or designs. Rather, use of the word exemplary is intended to present concepts in a concrete fashion. As used in this 5 application, the term "or" is intended to mean an inclusive "or" rather than an exclusive "or". That is, unless specified otherwise, or clear from context, "X employs A or B" is intended to mean any of the natural inclusive permutations. That is, if X employs A; X employs B; or X employs both A and B, then "X employs A or B" is satisfied under any of the foregoing instances. In addition, the articles "a" and "an" as used in this application 10 and the appended claims may generally be construed to mean "one or more" unless specified otherwise or clear from context to be directed to a singular form. Also, at least one of A and B and/or the like generally means A or B or both A and B.

[0051] Also, although the disclosure has been shown and described with respect to one or more implementations, equivalent alterations and modifications will occur to others

15 skilled in the art based upon a reading and understanding of this specification and the annexed drawings. The disclosure includes all such modifications and alterations and is limited only by the scope of the following claims. In particular regard to the various functions performed by the above described components (e.g., elements, resources, etc.), the terms used to describe such components are intended to correspond, unless otherwise

20 indicated, to any component which performs the specified function of the described component (e.g., that is functionally equivalent), even though not structurally equivalent to the disclosed structure which performs the function in the herein illustrated exemplary implementations of the disclosure. In addition, while a particular feature of the disclosure may have been disclosed with respect to only one of several implementations, such feature

25 may be combined with one or more other features of the other implementations as may be desired and advantageous for any given or particular application. Furthermore, to the extent that the terms "includes", "having", "has", "with", or variants thereof are used in either the detailed description or the claims, such terms are intended to be inclusive in a manner similar to the term "comprising."

What is claimed is:

1. A method for providing a client application with access to a file, comprising:

receiving a request from a client application to access a file;

5 if a local offline version of the file is stored locally, then providing the client application with access to the local offline version as the file; and

if a web online version of the file is stored remotely, then providing the client application with access to the web online version of the file as the file.

2. The method of claim 1, the providing the client application with access to the local 10 offline version comprising providing the client application with a local file path associated with the local offline version of the file, and the providing the client application with access to the web online version comprising providing the client application with a URL associated with the web online version.

3. The method of claim 1, comprising:

15 providing the client application with access to both the local offline version of the file and the web online version of the file as the file.

4. The method of claim 3, comprising:

synchronizing the file, the local offline version of the file, and the web online version of the file while the client application accesses the file.

20 5. The method of claim 3, comprising:

enabling co-authoring of the file between the client application and a remote client application accessing a corresponding file, the file and the corresponding file synchronized through the web online version of the file.

6. The method of claim 5, comprising:

25 providing a notification to the client application of a modification to the web online version of the file; and

instructing the client application to synchronize the file and the web online version of the file.

7. The method of claim 1, comprising:

30 providing the client application with a local file path associated with the local offline version of the file; and

delegating synchronization authorization to the client application to synchronize the file and the local offline version of the file using the local file path.

8. The method of claim 1, comprising:
providing the client application with a URL associated with the web online version of the file; and

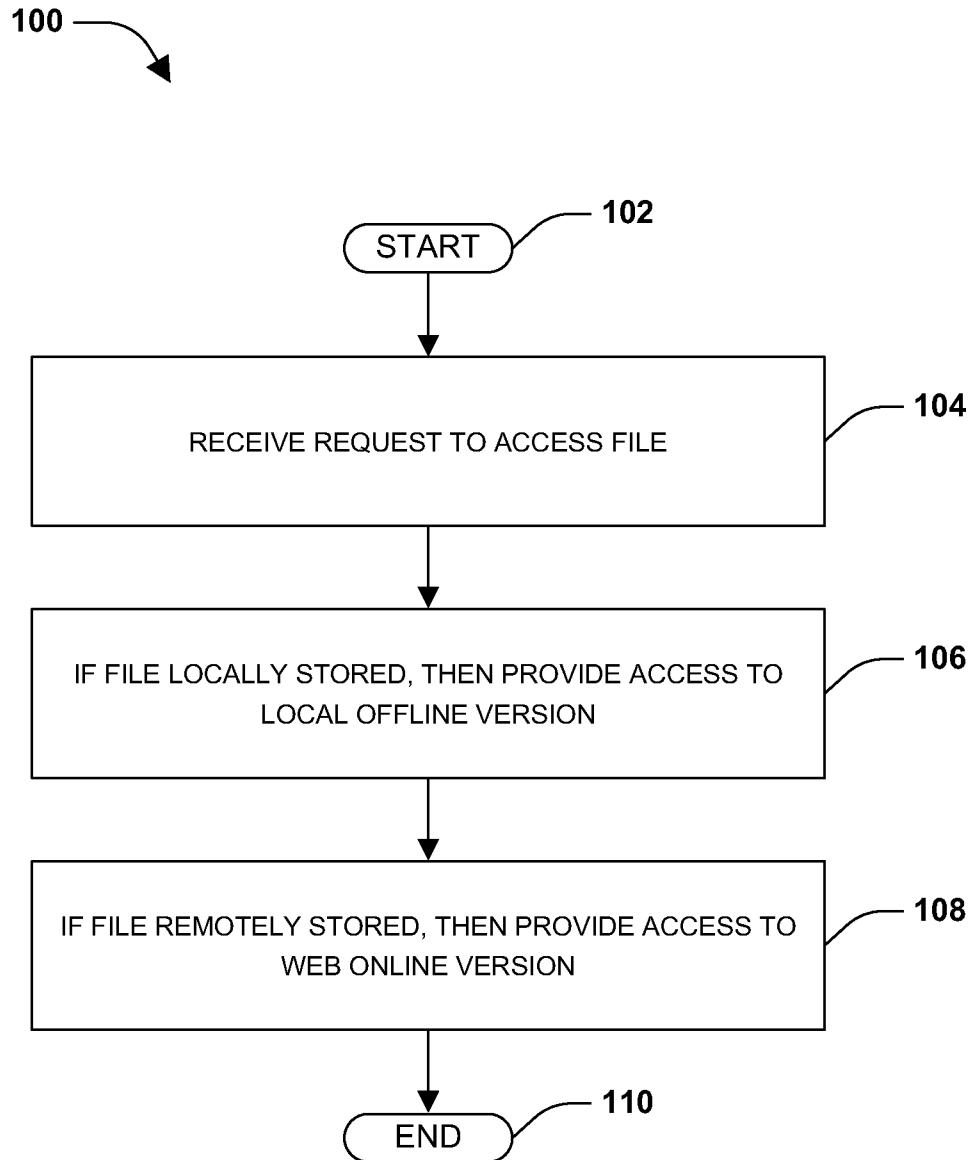
5 delegating synchronization authorization to the client application to synchronize the file and the web online version of the file using the URL.

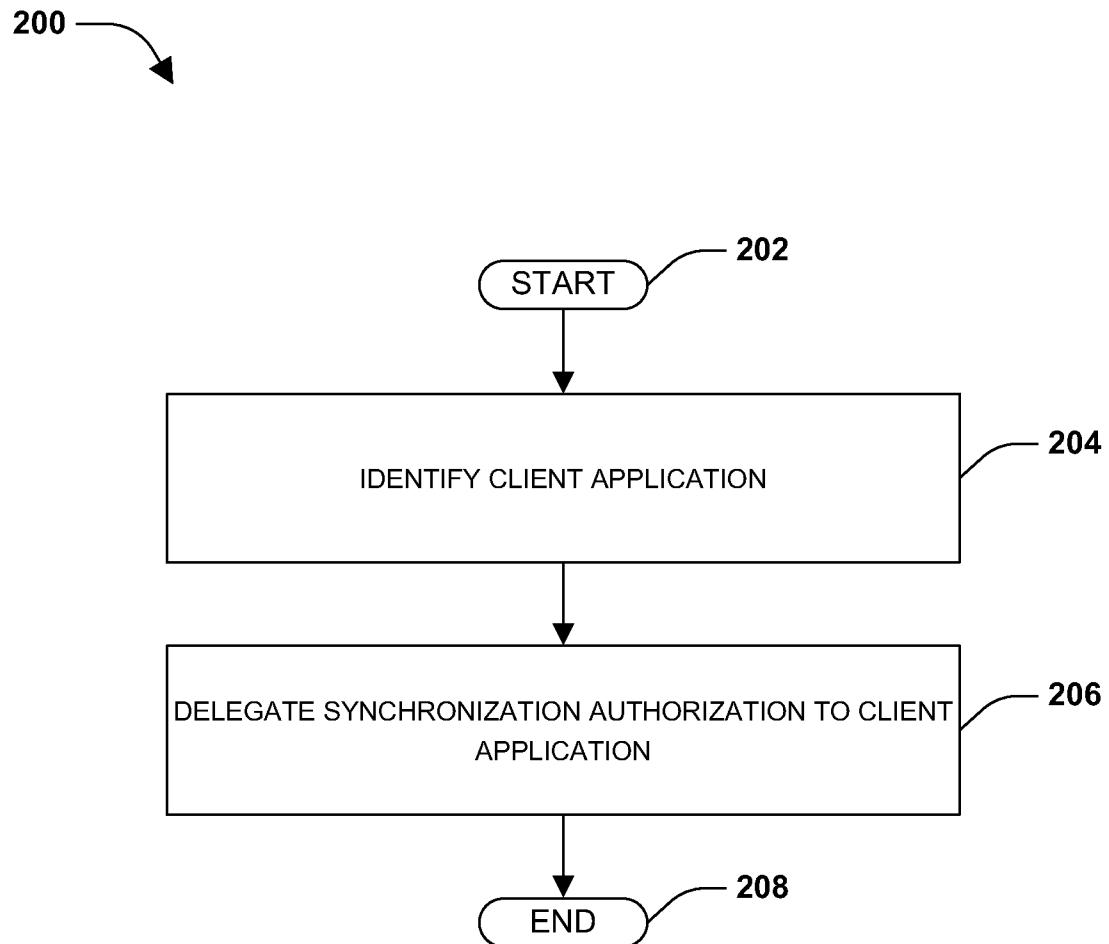
9. A system for providing a client application with offline access to a file, comprising:

a sync module configured to:

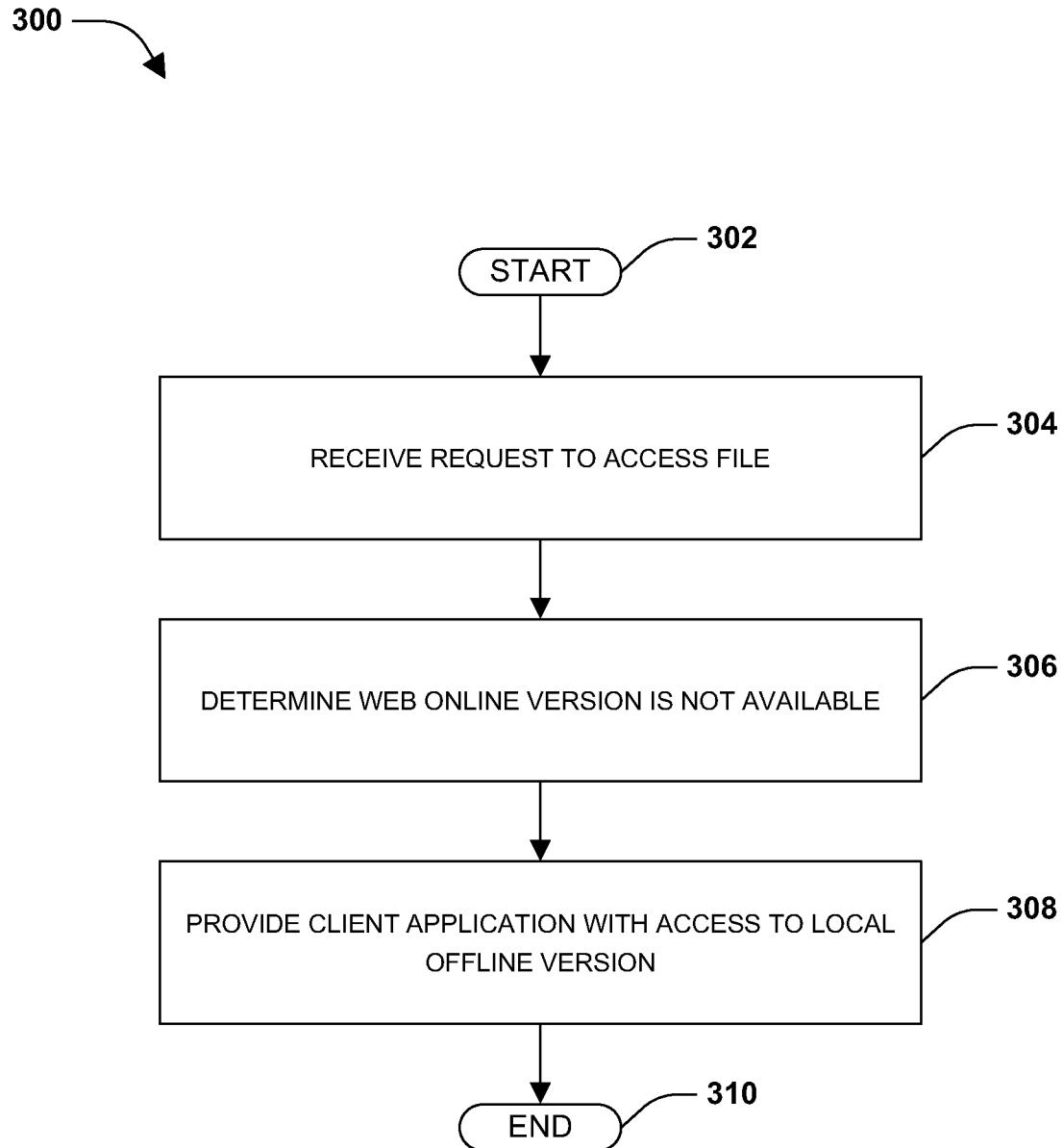
10 receive a request from a client application to access a file;
determine a web online version of the file is not available; and
provide the client application with access to a local offline version of the file.

10. The system of claim 9, the request comprising a URL associated with the web
15 online version of the file, and the sync module configured to provide the client application with a local file path associated with the local offline version of the file, the local file path determined based upon the URL.

**FIG. 1**

**FIG. 2**

3/9

**FIG. 3**

4/9

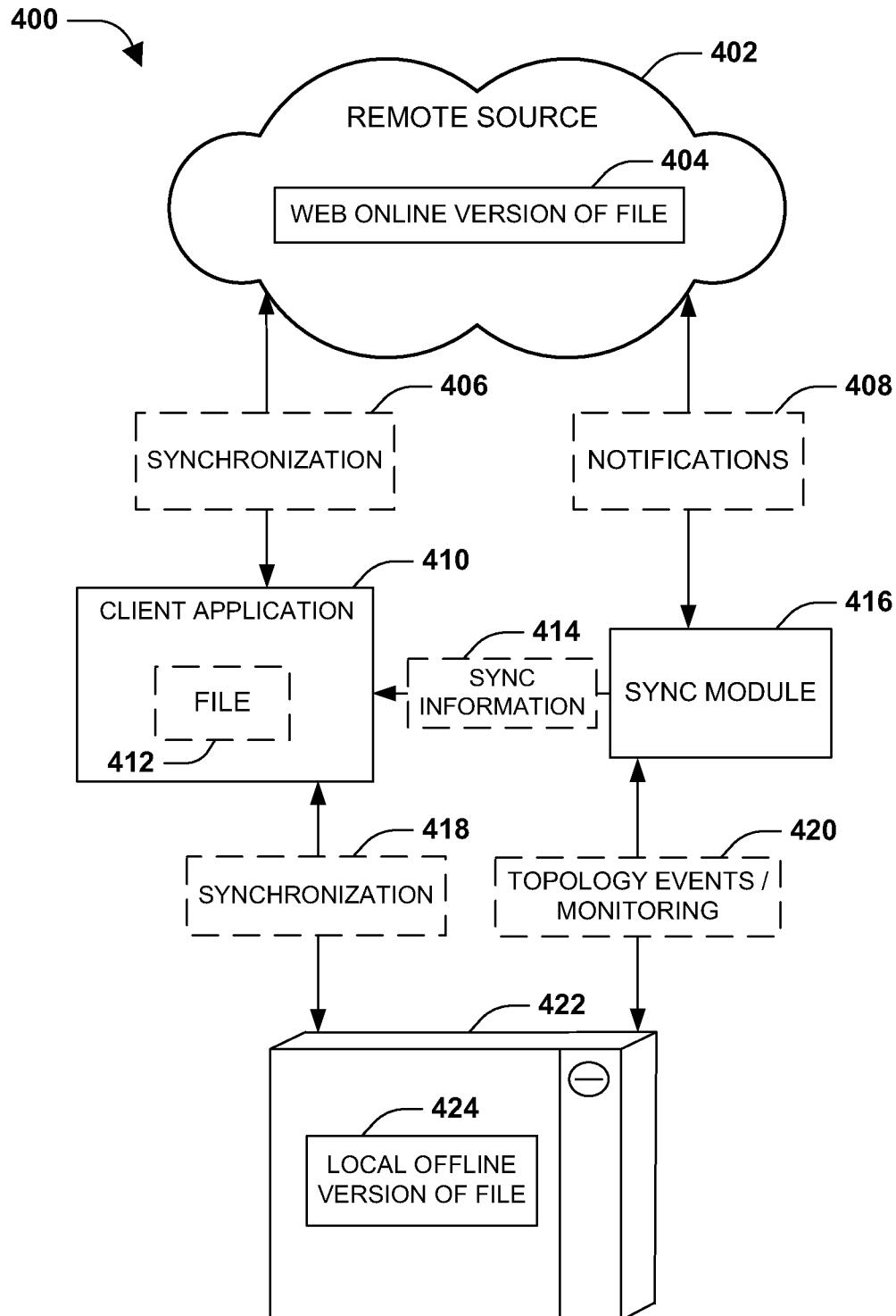
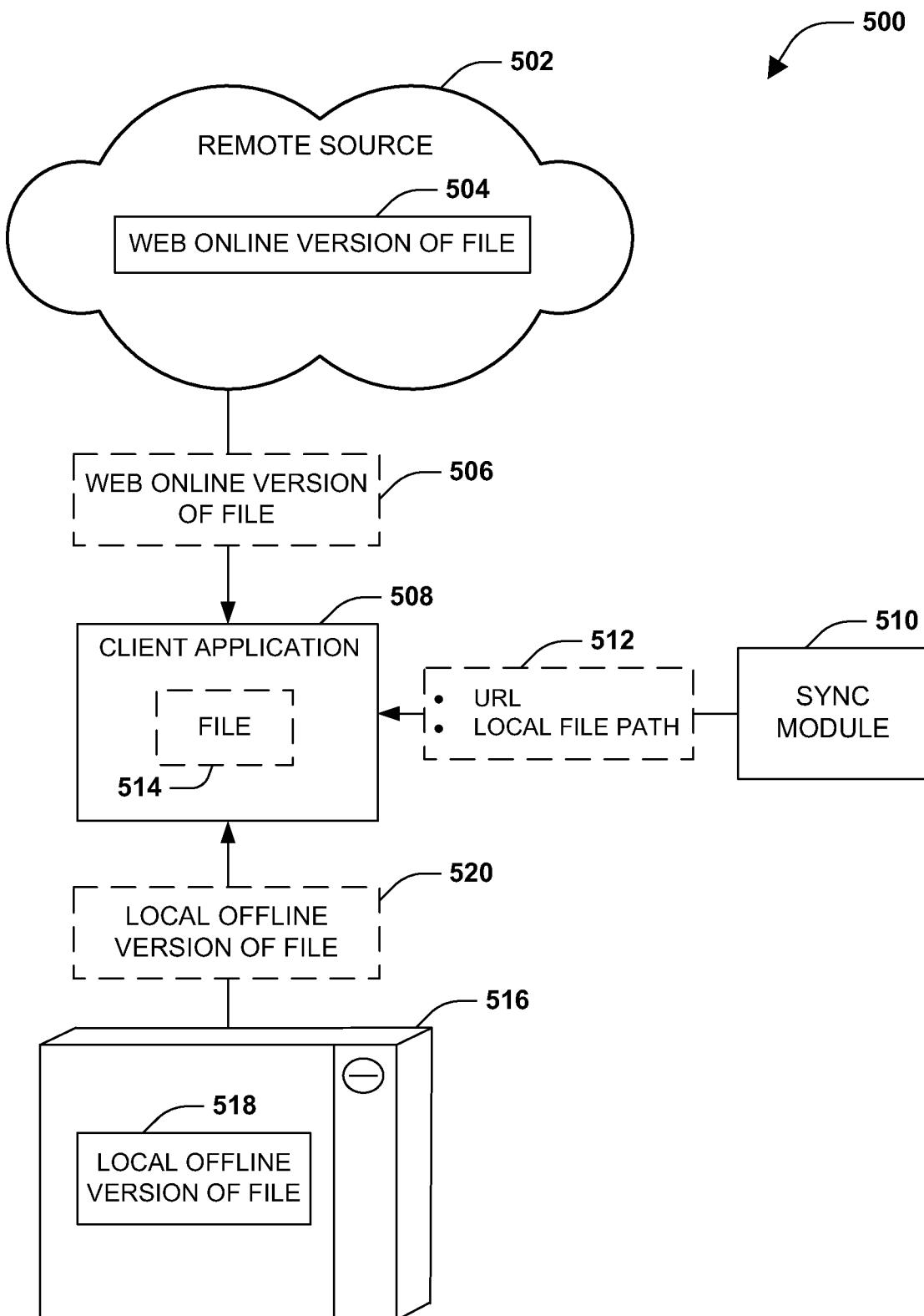
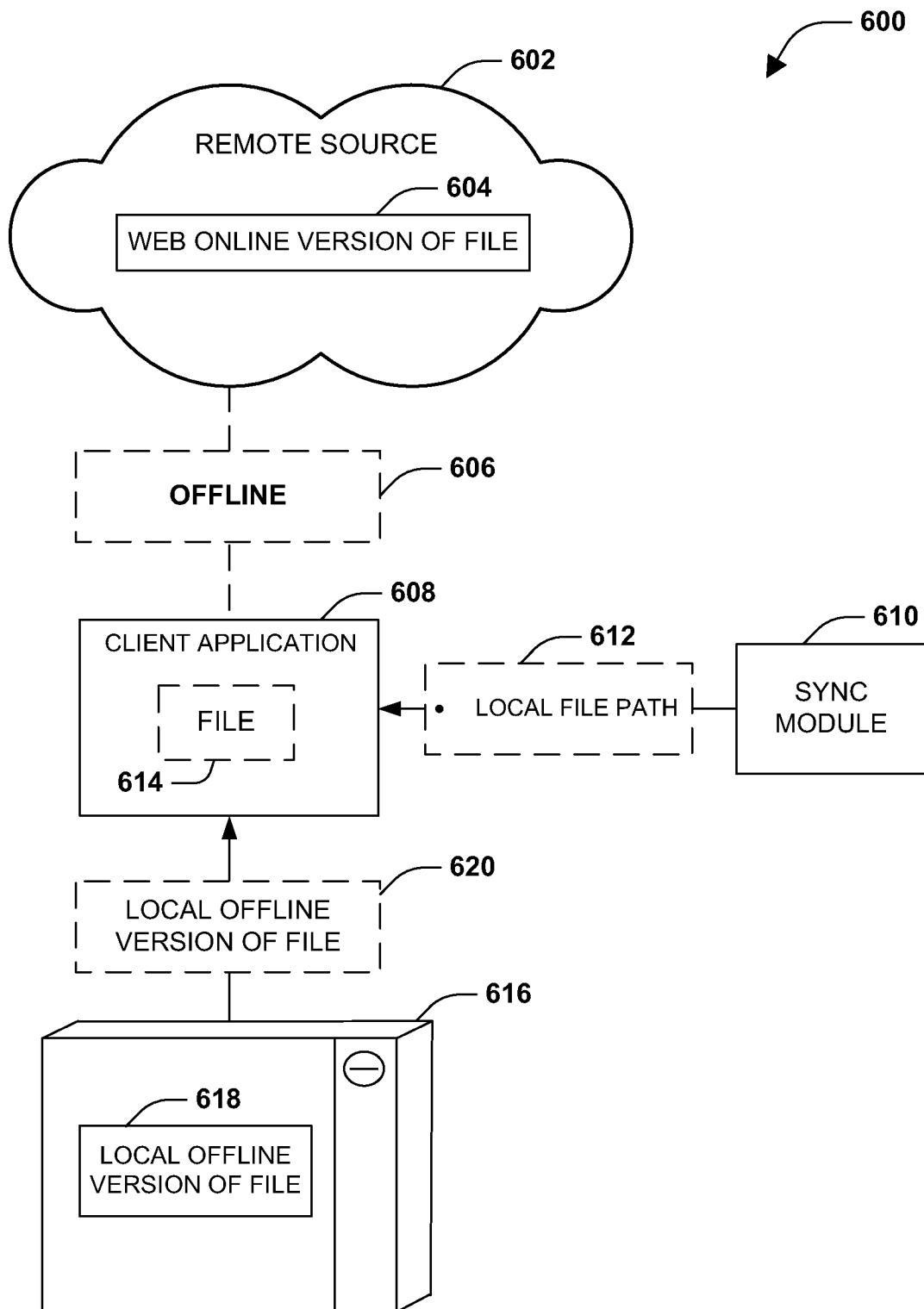


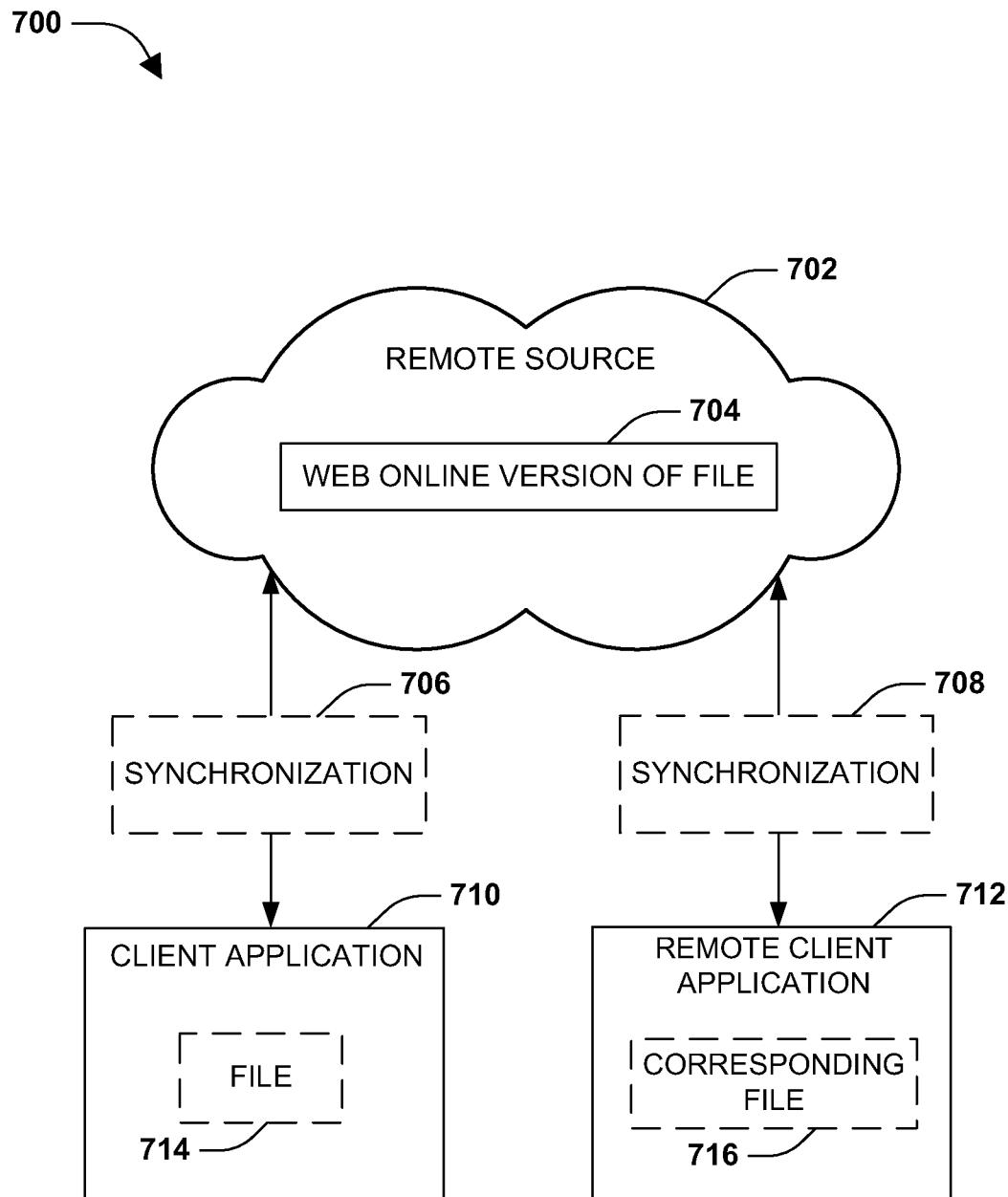
FIG. 4

5/9

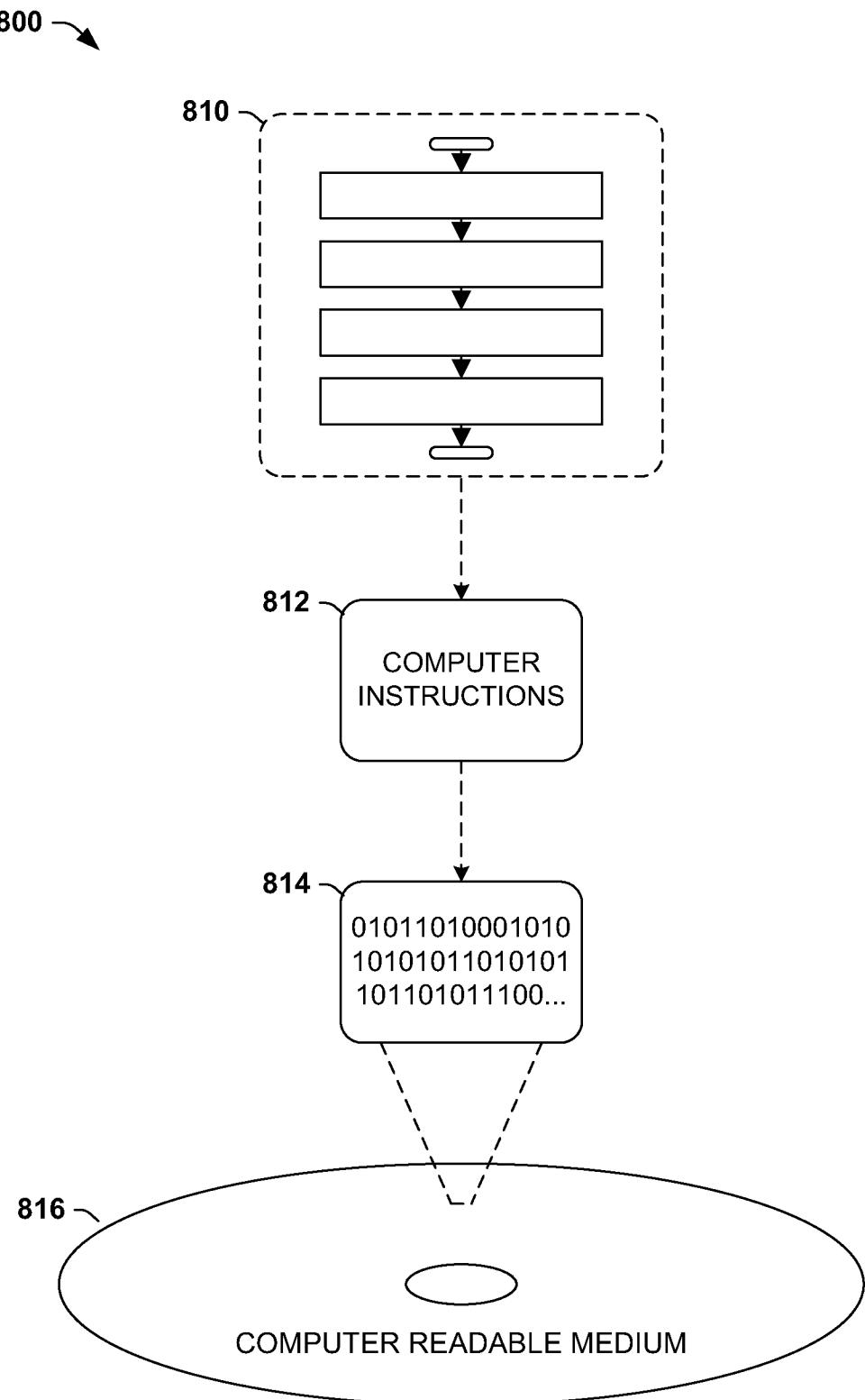
**FIG. 5**

**FIG. 6**

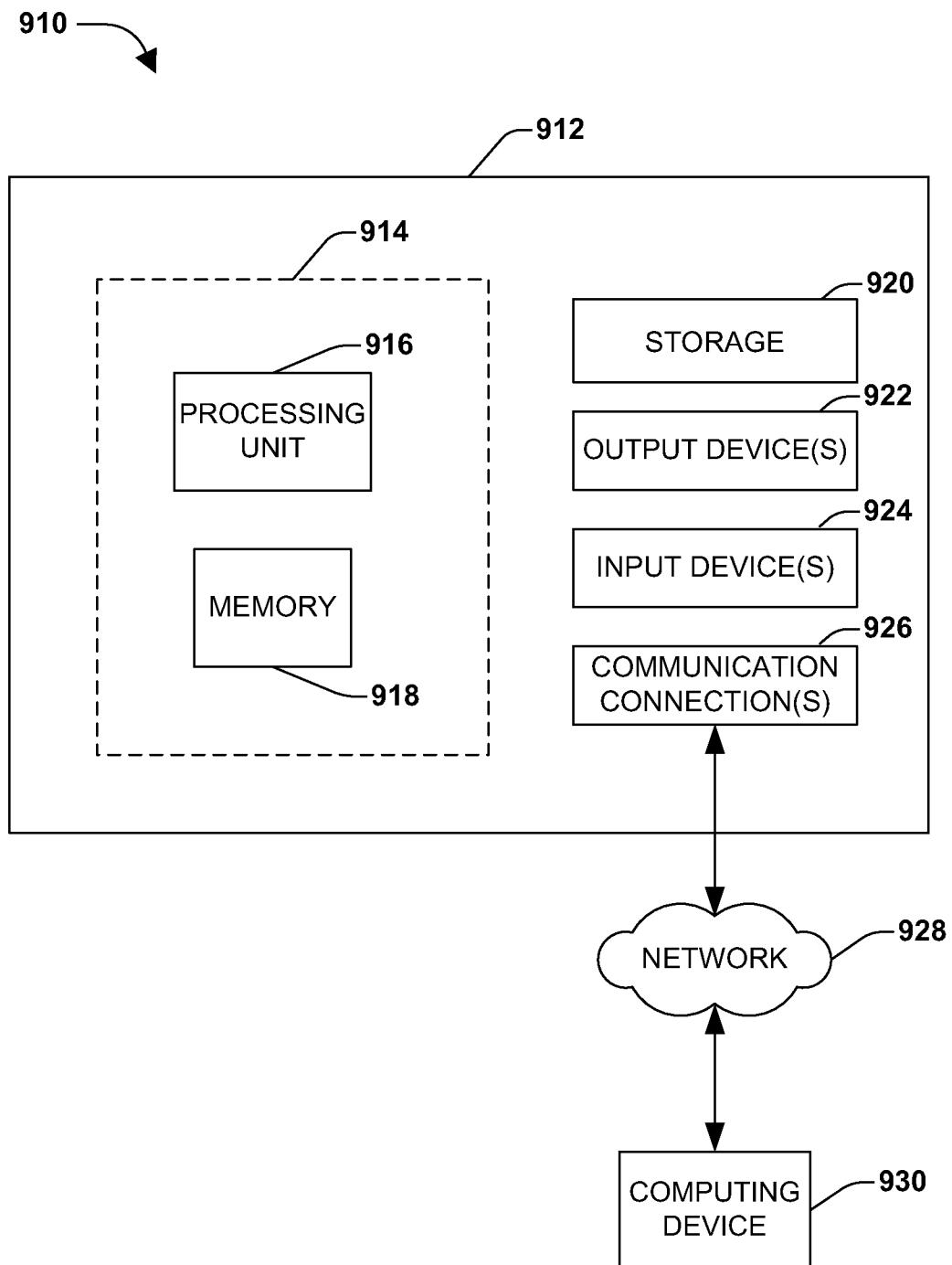
7/9

**FIG. 7**

8/9

**FIG. 8**

9/9

**FIG. 9**

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2012/065468

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009-0157627 A1	18.06.2009	CN 101861576 A CN 101861578 A CN 101861580 A CN 101861581 A CN 101861582 A EP 2203844 A2 EP 2206048 A2 EP 2206049 A2 EP 2206050 A2 EP 2206051 A2 KR 10-2010-0068473 A KR 10-2010-0080822 A KR 10-2010-0080824 A KR 10-2010-0080825 A KR 10-2010-0091165 A KR 10-2010-0105544 A US 07917584 B2 US 08099671 B2 US 08108426 B2 US 08112460 B2 US 08156146 B2 US 08234315 B2 US 08239511 B2 US 08280925 B2 US 2009-0157627 A1 US 2009-0157628 A1 US 2009-0158142 A1 US 2009-0164592 A1 US 2009-0171974 A1 US 2009-0171993 A1 US 2009-0172078 A1 US 2009-0172085 A1 US 2009-0172086 A1 US 2009-0172087 A1 US 2009-0172101 A1 US 2009-0172568 A1 US 2009-0172569 A1 US 2009-0172702 A1 US 2009-0172715 A1 US 2009-0175198 A1 US 2009-0177734 A1 US 2009-0192969 A1 US 2009-0192992 A1 US 2009-0193410 A1 US 2009-0193440 A1 US 2009-0254610 A1 WO 2009-043029 A2 WO 2009-043029 A3 WO 2009-043029 A4	13.10.2010 13.10.2010 13.10.2010 13.10.2010 13.10.2010 07.07.2010 14.07.2010 14.07.2010 14.07.2010 14.07.2010 23.06.2010 12.07.2010 12.07.2010 12.07.2010 18.08.2010 29.09.2010 29.03.2011 17.01.2012 31.01.2012 07.02.2012 10.04.2012 31.07.2012 07.08.2012 02.10.2012 18.06.2009 18.06.2009 18.06.2009 25.06.2009 02.07.2009 02.07.2009 02.07.2009 02.07.2009 02.07.2009 02.07.2009 02.07.2009 02.07.2009 02.07.2009 02.07.2009 02.07.2009 09.07.2009 09.07.2009 30.07.2009 30.07.2009 30.07.2009 30.07.2009 08.10.2009 02.04.2009 23.07.2009 17.09.2009

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2012/065468

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		WO 2009-043030 A2 WO 2009-043030 A3 WO 2009-043030 A4 WO 2009-043033 A2 WO 2009-043033 A3 WO 2009-043033 A4 WO 2009-043035 A2 WO 2009-043035 A3 WO 2009-043035 A4 WO 2009-043037 A2 WO 2009-043037 A3 WO 2009-043037 A4 WO 2009-055499 A2 WO 2009-055499 A3 WO 2009-055499 A4	02.04.2009 04.06.2009 30.07.2009 02.04.2009 23.07.2009 17.09.2009 02.04.2009 25.06.2009 27.08.2009 02.04.2009 28.05.2009 30.07.2009 30.04.2009 16.07.2009 17.09.2009
KR 10-2011-0123867 A	16.11.2011	None	
US 2006-0242259 A1	26.10.2006	AU 2006-238811 A1 AU 2006-238811 B2 BR P10609105 A2 CA 2601892 A1 CN 101529437 A EP 1872247 A2 EP 1872247 A4 IL 185859 D0 JP 2008-538843 A JP 5094709 B2 KR 10-1238537 B1 KR 10-2008-0005486 A MX 2007012420 A NO 20074719 A NO 20074719 B NZ 561778 A RU 2007138964 A RU 2421795 C2 US 7607582 B2 WO 2006-116368 A2 WO 2006-116368 A3 ZA 200707904 A	02.11.2006 19.01.2012 17.02.2010 02.11.2006 09.09.2009 02.01.2008 17.11.2010 06.01.2008 06.11.2008 12.12.2012 28.02.2013 14.01.2008 19.10.2007 19.11.2007 19.11.2007 29.10.2010 27.04.2009 20.06.2011 27.10.2009 02.11.2006 16.04.2009 29.07.2009
US 2005-0027673 A1	03.02.2005	JP 2005-044104 A	17.02.2005