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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 November 2007 (15.11.2007)

PCT

(10) International Publication Number
WO 2007/129143 A2

(51) **International Patent Classification:**
H04L 12/58 **(2006.01)**

(21) **International Application Number:** **PCT/IB2006/004204**

(22) **International Filing Date:** 11 December 2006 (11.12.2006)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
60/748,988 9 December 2005 (09.12.2005) US

(71) **Applicant** *(for all designated States except US):* **EBUDDY HOLDING B.V.** [NL/NL]; Keizergracht 560-4, NL-1017 EM Amsterdam (NL).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only):* **TAYLOR, Paulo** [PT/NL]; Tweede Weteringdwarsstraat 62, NL-1017 SZ Amsterdam (NL). **RUEB, Jan-Joost** [NL/NL]; Eerste C Huygensstraat 102-3, NL-1054 BZ Amsterdam (NL). **BARKER, Onno** [NL/NL]; Van Ostadestraat 189 Sous, NL-1073 TM Amsterdam (NL).

(74) **Agent: DENDORFER, Claus;** Dendorfer & Herrmann, Bayerstr. 3, 80335 Munchen (DE).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available):* AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, 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, LV, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
- *without international search report and to be republished upon receipt of that report*
- *the filing date of the international application is within two months from the date of expiration of the priority period*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 2007/129143 A2

(54) **Title:** MESSAGE HISTORY DISPLAY SYSTEM AND METHOD

(57) **Abstract:** A technique for message history display includes combining message histories for multiple different messaging services. A system constructed according to the technique may include, for example, a message history database; a history aggregation engine that aggregates message logs for storage in the message history database; and a history provisioning engine that provides an aggregated message log associated with the user from the message history database to a requesting device. A method according to the technique may include, for example, identifying a device in association with a user profile; providing an online platform that receives messages from and sends messages to the device; and creating an aggregated log from messages sent to and from the device.

MESSAGE HISTORY DISPLAY SYSTEM AND METHOD

BACKGROUND

Instant messaging requires the use of a client program that hooks up an instant messaging service and differs from e-mail in that conversations are then able to happen in real time. Most services offer a presence information feature, indicating whether people on one's list of contacts are currently online and available to chat. This may be called a contact list. In early instant messaging programs, each letter appeared as it was typed, and when letters were deleted to correct typos this was also seen in real time. This made it more like a telephone conversation than exchanging letters. In modern instant messaging programs, the other party in the conversation generally only sees each line of text right after a new line is started. Most instant messaging applications also include the ability to set a status message, roughly analogous to the message on a telephone answering machine.

Popular instant messaging services on the public Internet include .NET Messenger Service, AOL Instant Messenger, Excite/Pal, Gadu-Gadu, Google Talk, iChat, ICQ, Jabber, Qnext, QQ, Meetro, Skype, Trillian and Yahoo! Messenger. These services owe many ideas to an older (and still popular) online chat medium known as Internet Relay Chat (IRC).

The foregoing examples of the related art and limitations related therewith are intended to be illustrative and not exclusive. Other limitations of the related art will become apparent to those of skill in the art upon a reading of the specification and a study of the drawings.

SUMMARY

The following embodiments and aspects thereof are described and illustrated in conjunction with systems, tools, and methods that are meant to be exemplary and illustrative, not limiting in scope. In various embodiments, one or more of the above-described problems have been reduced or eliminated, while other embodiments are directed to other improvements.

A technique for message history display includes combining message histories for multiple different messaging services. A system constructed according to the technique may include, for example, a message history database; a history aggregation engine that aggregates the message logs for storage in the message history database; and a history provisioning engine that provides an aggregated message log associated with the user from the message history database to a requesting device.

A method according to the technique may include, for example, identifying a device in association with a user profile; providing an online platform that receives messages from and sends messages to the device; and creating an aggregated log from messages sent to and from the device.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the inventions are illustrated in the figures. However, the embodiments and figures are illustrative rather than limiting; they provide examples of the invention.

FIG. 1 depicts an example of a system for providing instant messages to clients via a web interface.

FIG. *2* depicts an example of a system for displaying content from an IM client at an alternative IM client.

FIG. 3 depicts an example of a system for message history aggregation and provisioning.

FIG. 4 depicts a flowchart of an example of a method for message log aggregation and provisioning.

FIG. -s depicts a flowchart of another example of a method for message log aggregation and provisioning.

FIG. 6 depicts a computer system suitable for implementation of the techniques described with reference to FIGS. 1-5, 7-8.

FIG. 7 depicts a flowchart of an example of a method for message log aggregation and provisioning.

FIG. 8 depicts a flowchart of another example of a method for message log aggregation and provisioning.

DETAILED DESCRIPTION

In the following description, several specific details are presented to provide a thorough understanding of embodiments of the invention. One skilled in the relevant art will recognize, however, that the invention can be practiced without one or more of the specific details, or in combination with other components, etc. In other instances, well-known implementations or operations are not shown or described in detail to avoid obscuring aspects of various embodiments, of the invention.

FIG. 1 depicts an example of a system 100 for providing instant messages to clients via a web interface. In the example of FIG. 1, the system 100 includes a network 102, a server 104, and an Instant Messenger (IM) server 106, and an IM network 108. The server 104 is coupled to the network at least by way of two way communications. The two way communication (e.g., via port 80) is represented in the example of FIG. 1 as an arrow 110. The server 104 is coupled to the IM server 106 via one or more other ports. The two way communication via the other ports is represented in the example of FIG. 1 as an arrow 112. The IM server 106 is coupled to the IM network 108 via any known or convenient mechanism. Indeed, the EvI server 106 may be thought of as part of the IM network 108. The network 102 couples a plurality of clients 114-1 to 114-N (referred to collectively as clients 114) to the server 104. In the example of FIG. 1, the server 104 includes an event queue 116.

The network 102 may include by way of example but not limitation 1,AN, WAN, VLAN, WLAN, Internet, cellular network, phone network, radio network, or some other known or convenient network. The term "Internet" as used herein refers to a network of networks that uses certain protocols, such as TCP/IP, and possibly other protocols such as the hypertext transfer protocol (HTTP) for hypertext markup language (HTML) documents that make up the World Wide Web (the web). The physical connections of the Internet and the protocols and communication procedures are well known, but any convenient physical connections or protocols could be used.

The server 104 may include a multiple servers. Indeed, it may be desirable, depending upon details of a particular implementation, to install several servers to cope with the number of simultaneous users the system 100 supports. It may further be desirable, depending upon details of a particular implementation, for the server 104 to have a high CPU throughput, together with large amounts of RAM, to handle a large number of users. It may further be desirable, depending upon details of a particular implementation, to accomplish resource sharing via thread handling where a pool of threads is shared and used by one or more of the clients 114 for client-server communication and between the server 104 and the IM server 106.

The server 104 may include one or more of an application server, database server, web server, banners server, and content server, or any combination thereof. To make the most of the techniques described herein, the server 104 should, though is not required to, include at least one application server. The other servers can have supporting roles in, by way of example but not limitation, serving static content or advertising (e.g., banners), storing usage data, or fulfilling some other known or convenient function.

The server 104 may act as a proxy server between the clients 114 and the HvI server 106. The server 104 receives communications from the clients 114 on http port 80, and responds to the clients 114 on http port 80. Communications from the clients 114 that are bound for the IM network 108, however, must also come through http port 80 to the server 104, and are then forwarded to the IM server 106. In this way, the server 104 acts as a carrier of the data from users to the IM network 108 using a mechanism that controls and manages the data (e.g., text messages, display images, emoticons, audio/video streams, etc.) sent between one of the clients 114 and the server 104, and vice versa.

The IM server 106 may be any known or convenient IM server that is compatible with IM. Events, messages, or other appropriate data from the IM server 106 are collected in the event queue 116 of the server 104. The events may be collected in association with a variety of protocols including by way of example but not limitation port 1863, port 5050_3port 5222, port 5190, etc.

The DvI network 108 may include one or a combination of networks selected from MSN Messenger, Yahoo! Messenger, AIM AOL, ICQ, QQ, Jabber, Google Talk, IRC, or some other known or convenient IM network.

The clients 114 may include any known or convenient device, including by way of example but not limitation, a Web browser, mobile client, PDA, game console, TV box, native application, etc. The clients poll the server 104 for events. The events can be removed from the event queue 116 and translated into text, JavaScript, XML, or some other known or convenient format that one or more of the clients 114 need or expect in order to process data associated with the event.

To interact with the JJvI network 108, the clients 114 send data to the server *104*. The data, which may include commands, is processed and translated into corresponding data that will be sent to the appropriate JJVI network. In an embodiment, the appropriate IM network may be determinable based upon the protocol encoded in a message.

Messages or actions from the clients 114 are collected over network protocols such as, by way of example but not limitation, HTTP or plain socket connections. The messages or actions are transformed to an appropriate protocol format to be sent over a compliant port from the clients 114 to the server 104, with the IM protocol on the application side. In a non-limiting embodiment, the compliant port is http port 80. However, any port having similar characteristics to those of a typical port 80 could be used.

The latest available browsers, as of December 2005, enable the use of a technique called AJAX (Asynchronous JavaScript And XML). With AJAX, appropriately configured clients 114 can execute actions and poll for messages or events using only JavaScript. The method is based on using an XMLHttpRequest object to make HTTP requests to the server 104. The server 104 may reply with messages taken from the queue of the corresponding session in XML (or another) format that are parsed and displayed according to the message content.

For clients 114 that include a browser, when accessing the server 104 the browser typically uses hidden HTML frames to update information on visible frames. The visible frames display appropriate information while the hidden frames are reloaded in short periods of time. In each refresh that hits the server 104, the browser identifies the current messaging session and checks if new events or messages associated with the session are in the event queue 116. When new information arrives and needs to be displayed in some form, the browser makes use of, for example, JavaScript code to update the visible frames and windows with new messages or events keeping the information up to date in the screen. In this way, automatic refreshing can take place in a hidden frame.

In another embodiment, certain of the clients 114 with browsers may not make use of refreshes. For example, a form of updating the screen without using a refresh technique is to keep one single HTTP socket request alive for the whole period of a messaging session without actually closing the socket connection. In this example, information is initially loaded and displayed in one single visible frame. While events and messages are being received by the server 104, JavaScript code can be injected into the HTML document through the same HTTP socket kept alive and managed by the server 104. For each event or message, the browser can interpret the JavaScript code injected and the corresponding parts of the HTML document and windows will be updated.

In another embodiment, certain of the clients 114 with browsers may make use of manual refreshes. Some relatively unsophisticated browsers, such as WAP and xHTML browsers often available on mobile phones, do not support hidden frames and/or JavaScript (and others may be configured such that they do not support hidden frames and/or JavaScript). In such cases, the information displayed has to be updated manually by the user. Manual updating enables any mobile phone, PDA, TV Set or any device with a browser to connect to the server 104 and use the messaging platforms made available by the server 104 assuring the communication between the clients 114 and the IM server 106.

Message history can be stored by most IM clients on a local computer. For alternative web and mobile-based clients local storage may not be possible. In a non-limiting embodiment, the server 104, may have the capability to store message history from IM conversations done via one or more of the clients 114. The message history can be accessed and searched at any time via the server 104 by one or more of the clients 114

FIG. 2 depicts an example of a system 200 for displaying content from an IM client at an alternative BvI client. In the example of FIG. 2, the system 200 includes a client 202, an IM network204, a server 206, an IM network 208, a client210, other IM networks 212-1 to 212-N (referred to collectively as other IM networks 212), and other clients 214-1 to 214-N (referred to collectively as other clients 214).

For illustrative purposes, it is assumed that the client 202 has content that is compatible with the TM network 204. However, the client 210 is capable of reading content formatted to be compatible with the IM network 208. Thus, in operation, the server 206 collects content from the client 202 (either through the TM network 204, as shown in FIG. 2, or directly from the client 202, such as is shown by way of example in FIG. 1). The server .206 then formats the content as appropriate for use on the IM network 208. Once the content is properly formatted, it can be made available to the client 210 (either through the IM network 208, as shown in FIG. 2, or directly to the client 210, such as is shown by way of example in FIG. 1). Depending upon the embodiment and/or implementation, the content may also be formatted as appropriate for one or more of the other IM networks 212, to be made available for one or more of the other clients 214.

In an embodiment, the server 206 can save the content in one or many formats. In this way, the client 202 could make content available in a first IM format, the server 206 could convert the content into a second IM format, and the server 206 can save the content in at least the second IM format. Thus, the client 210 could receive the data in the second IM format. The server 206 could easily store the content in the first IM format, as well, and make the content available to other clients coupled to the IM network 204. In addition, the server 206 could convert the content to other IM formats, such as those formats that are associated with the other IM networks 212, and save the other IM formats. In this way, the other clients 214 may have access to the content.

The client 202 and the client 210 (and the clients 214) may or may not be associated with a single user. Advantageously, when a single user is associated with more than one client, the user can view content at any one client normally available on some other type of network. Thus, for example,- a user who uses a cell phone for chat and an IM client for chat can view an aggregation of a mobile phone chat log and an IM chat history on the cell phone or on the IM client.

FIG. 3 depicts an example of a system 300 for message history aggregation and provisioning. The system 300 includes user devices 302, network 304, a server 306, a user profile database 308, and a message history database 310.

The user devices 302 may include a mobile phone 312, an DvI client 322, a web browser 332, and other client 342. The networks 304 may include a cellular network 314 coupled to the mobile phone 312, an IM network 324 coupled to the IM client 322, a web network 334 coupled to the web browser 332, and other network 344 coupled to the other client 342. Each of the networks 304 is coupled to the server 306. It may be noted that the other client 342 may include another mobile phone, IM client, web browser, or some other known or convenient client. Similarly, the other network 344 may include a cellular network, BVI network, web network, or some other known or convenient network.

In the example of FIG. 3, the server 306 includes a history aggregation engine 316, a user profile database interface 318, a message history database interface 320, and a history provisioning engine 326. The user profile database interface 318 and the message history database interface .320 may be incorporated into a single database interface, depending upon the implementation.

The history aggregation engine 316, which maybe embodied in a computer-readable medium at the server 306, is for taking message histories from a plurality of environments associated with a single user identified in the user profile database 308 and aggregating the message histories in the message history database 310.

For example, a mobile phone chat log associated with the mobile phone 312 may be stored on the mobile phone 312, in the cellular network 314, or elsewhere. The history aggregation engine 316 can access the user profile database 308 through the user profile database interface 318 to determine where the mobile phone chat log is stored (or, for example, the mobile phone 312 could be queried). The mobile phone chat log can be stored in its current format in the message history database 310, or the mobile phone chat log could be converted to a generic format or some other format (e.g., an XML format) prior to storage in the message history database 310. Regardless of the format chosen by implementation or configuration, the history aggregation engine 316 can provide some or all of the mobile phone chat log to the message history database interface 320 for storage in the message history database 310.

As another example, an IM chat history associated with the IM client 314 may be stored on the IM client 322, in the IM network 324, or elsewhere (e.g., on a high level IM network server). As with the mobile phone chat log example, the history aggregation engine 316 can access the user profile database 308 through the user profile database interface 318 to determine where the IM chat history is stored and convert some or all of the IM chat history, if necessary, for storage in the message history database 310.

As another example, a web-based chat history associated with the web browser 332 may be stored on a computer on which the web browser 332 is embodied in a computer-readable medium, in the web network 334, or elsewhere. The history aggregation engine 316 can access the user profile database 308 through the user profile database interface 318 to determine where the web-based chat history is stored and convert some or all of the web-based chat history, if necessary, for storage in the message history database 310. In one implementation, all of the other histories are converted into a web-based format, and aggregated for a user for display on the web browser 332. Of course, in other implementations, other formats (or a generic format so that display is facilitated on any of the user's devices) can be used.

The history provisioning engine 326, which may be embodied in a computer-readable medium at the server 306, may take a request for a log from one of the user devices 302 and query the message history database interface 320 for an aggregated log stored in the message history database 310. Depending upon the implementation, the history provisioning engine 326 may receive a plurality of logs in different formats associated with each of the user devices 302 and convert the different formats into a format suitable for display on the device from which a request for a log was received; receive a plurality of logs, each of which is in a generic format, and convert the generic format into a format suitable for display on the device from which a request for a log was received; or receive a plurality of logs, each of which is in a particular format (e.g., XML) and convert the particular format into a format suitable for display on the device from which a request for a log was received, if the device is not suited to display the particular format. Alternatively, the message history database 310 could include multiple formats for the same log, and the format most suitable for provisioning to a requesting device may be provided to the history provisioning engine 326, which, in this case, would not need to reformat the log. In any case, the history provisioning engine 326 provides the aggregated log in a suitable format to a requesting device.

FIG. 4 depicts a flowchart 400 of an example of a method for message log aggregation and provisioning. This method and other methods are depicted as serially arranged modules. However, modules of the methods may be reordered, or arranged for parallel execution as appropriate. In the example of FIG. 4, the flowchart 400 begins at module 402 where a plurality of devices are identified in association with a user profile.

In the example of FIG. 4, the flowchart 400 continues to module 404 where a plurality of message logs respectively associated with the plurality of devices are stored. The message logs may or may not be stored on separate devices. For example, an IM chat history may be stored on an IM server, while a mobile phone chat log may be stored on the mobile phone.

In the example of FIG. 4, the. flowchart 400 continues to module 406 where a request for a message log of the plurality of message logs is received. Such a request would typically (though not necessarily) be received from a device that uses the messaging protocol that results in the message log. In an alternative, the device may send a request explicitly for the aggregated message log. In such an alternative, a response to a request for the (unaggregated) message log may include the message log and not the aggregated message log.

In the example of FIG. 4, the flowchart 400 continues to module 408 where the plurality of message logs are compiled into an aggregated message log having a format suitable for display on a device associated with the requested message log. The aggregated message log may be compiled on the fly (e.g., when the request is received) or could be stored in advance in anticipation of the Tequest. If stored in advance, multiple redundant aggregated message logs could be stored in different formats, each of which is suitable for display on a device associated with the user profile.

In the example of FIG. 4, the flowchart 400 continues to module 410 where the aggregated message log is provided in response to the request. The aggregated message log would typically (though not necessarily) be sent to the device that uses the messaging protocol that results in the requested message log. In a non-limiting embodiment, the device need not request the aggregated message, but receives it anyway.

In the example of FIG. 4, the various message logs are kept in their native formats. The message logs could reside in known or convenient locations associated with the various formats (e.g., a mobile phone chat log could be stored on the mobile phone). When compiling the various message logs into an aggregated message log, each of the locations may need to be checked for data. This can take some time. Also, some devices may be off (e.g., a mobile phone could be off) or off-line (e.g., an IM server that includes an IM chat log could be down for maintenance), which could result in an incomplete aggregated message log. This problem can be ameliorated by storing the aggregated message log in each possible requested format for a particular user, though this solution may result in wasting storage resources.

FIG. 5 depicts a flowchart 500 of another example of a method for message log aggregation and provisioning. In the example of FIG. 5, various message logs are stored in an aggregated log format. In the example of FIG. 5, the flowchart 500 starts at module 502 where a plurality of devices are identified in association with a user profile.

In the example of FIG. 5, the flowchart 500 continues to module 504 where a plurality of message logs respectively associated with the plurality of devices are converted into an aggregated log format. The aggregated log format may be the same format as would be suitable for display on at least one of the devices associated with a user, or the aggregated log format may be a generic format that would not be displayed on any of a user's devices without reformatting.

In the example of FIG. 5, the flowchart 500 continues to module 506 where the plurality of message logs are stored in the aggregated log format. Advantageously, the plurality of message logs can be (though are not necessarily) collected and stored on a single device or collection of devices that are relatively local with respect to one another. This may improve the speed with which the aggregated message log can be accessed.

In the example of FIG. 5, the flowchart 500 continues to module 508 where a request is received from a device of the plurality of devices for a message log of the plurality of message logs. In an alternative, the request may be explicitly for the aggregated message log. In this alternative, it may be possible to request and receive a non-aggregated message log, as well.

In the example of FIG. 5, the flowchart 500 continues to module 510 where the plurality of message logs are converted from the aggregated log format to a format associated with the requested message log. This conversion may or may not be necessary, depending upon the implementation. For example, the aggregated format may be a format that is not suitable for display without conversion on any of the devices associated with the user profile. As another example, the aggregated format may be a format that is suitable for display on one or more of the devices associated with the user profile (making conversion unnecessary for a subset of the devices), but unsuitable for display on one or more other devices associated with the user profile (making conversion necessary for a subset of the devices).

In the example of FIG. 5, the flowchart 500 continues to module 512 where the plurality of message logs are provided in the format associated with the requested message log to the requesting device. Advantageously, since the format of all of the message logs is converted (if necessary) to a format that is suitable for display on the requesting device, no matter from which device a user requests a message history, the aggregated message history can be displayed.

FIG. 6 depicts a computer system 600 suitable for implementation of the techniques described above with reference to FIGS. 1-5 (and 7-8, described later). The computer system 600 includes a computer 602, I/O devices 604, and a display device 606. The computer 602 includes a processor 608, a communications interface 610, memory 612, display controller 614, non-volatile storage 616, and I/O controller 618. The computer 602 may be coupled to or include the I/O devices 604 and display device 606.

The computer 602 interfaces to external systems through the communications interface 610, which may include a modem or network interface. The communications interface 610 can be considered to be part of the computer system 600 or apart of the computer 602. The communications interface 610 can be an analog modem, ISDN modem, cable modem, token ring interface, satellite transmission interface (e.g. "direct PC"), or other interfaces for coupling a computer system to other computer systems. Although conventional computers typically include a communications interface of some type, it is possible to create a computer that does not include one, thereby making the communications interface 610 optional in the strictest sense of the word.

The processor 608 may include, by way of example but not limitation, a conventional microprocessor such as an Intel Pentium microprocessor or Motorola power PC microprocessor. While the processor 608 is a critical component of all conventional computers, any applicable known or convenient processor could be used for the purposes of implementing the techniques described herein. The memory 612 is coupled to the processor 608 by a bus 620. The memory 612, which may be referred to as "primary memory," can include Dynamic Random Access Memory (DRAM) and can also include Static RAM (SRAM). The bus 620 couples the processor 608 to the memory 612, and also to the non-volatile storage 616, to the display controller 614, and to the I/O controller 618.

The I/O devices 604 can include a keyboard, disk drives, printers, a scanner, and other input and output devices, including a mouse or other pointing device. For illustrative purposes, at least one of the I/O devices is assumed to be a block-based media device, such as a DVD player. The display controller 614 may control, in a known or convenient manner, a display on the display device 606, which can be, for example, a cathode ray tube (CRT) or liquid crystal display (LCD).

The display controller 614 and I/O controller 618 may include device drivers. A device driver is a specific type of computer software developed to allow interaction with hardware devices. Typically this constitutes an interface for communicating with the device, through a bus or communications subsystem that the hardware is connected to, providing commands to and/or receiving data from the device, and on the other end, the requisite interfaces to the OS and software applications.

The device driver may include a hardware-dependent computer program that is also OS-specific. The computer program enables another program, typically an OS or applications software package or computer program running under the OS kernel, to interact transparently with a hardware device, and usually provides the requisite interrupt handling necessary for any necessary asynchronous time-dependent hardware interfacing needs.

The non-volatile storage 616, which may be referred to as "secondary memory," is often a magnetic hard disk, an optical disk, or another form of storage for large amounts of data. Some of this data is often written, by a direct memory access process, into memory 612 during execution of software in the computer 602. The non-volatile storage 616 may include a block-based media device. The terms "machine-readable medium" or "computer-readable medium" include any known or convenient storage device that is accessible by the processor 508 and also encompasses a carrier wave that encodes a data signal.

The computer system 600 is one example of many possible computer systems which have different architectures. For example, personal computers based on an Intel microprocessor often have multiple buses, one of which can be an *I/O* bus for the peripherals and one that directly connects the processor 608 and the memory 612 (often referred to as a memory bus). The buses are connected together through bridge components that perform any necessary translation due to differing bus protocols.

Network computers are another type of computer system that can be used in conjunction with the teachings provided herein. Network computers do not usually include a hard disk or other mass storage, and the executable programs are loaded from a network connection into the memory 612 for execution by the processor 608. A Web TV system, which is known in the art, is also considered to be a computer system, but it may lack some of the features shown in FIG. *6,* such as certain input or output devices. A typical computer system will usually include at least a processor, memory, and a bus coupling the memory to the processor.

The computer system 600 may be controlled by an operating system (OS). An OS is a software program—used on most, but not all, computer systems—that manages the hardware and software resources of a computer. Typically, the OS performs basic tasks such as controlling and allocating memory, prioritizing system requests, controlling input and output devices, facilitating networking, and managing files. Examples of operating systems for personal computers include Microsoft Windows®, Linux, and Mac OS®. Delineating between the OS and application software is sometimes rather difficult. Fortunately, delineation is not necessary to understand the techniques described herein, since any reasonable delineation should suffice.

The lowest level of an OS may be its kernel. The kernel is typically the first layer of software loaded into memory when a system boots or starts up. The kernel provides access to various common core services to other system and application programs.

As used herein, algorithmic descriptions and symbolic representations of operations on data bits within a computer memory are believed to most effectively convey the techniques to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of operations leading to a desired result. The operations are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

An apparatus for performing techniques described herein may be specially constructed for the required purposes, or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, by way of example but not limitation, read-only memories (ROMs), RAMs, EPROMs, EEPROMs, magnetic or optical cards, any type of disk including floppy disks, optical disks, CD-ROMs, DVDs, and magnetic-optical disks, or any known or convenient type of media suitable for storing electronic instructions.

The algorithms and displays presented herein are not inherently related to any particular computer architecture. The techniques may be implemented using any known or convenient programming language, whether high level (e.g., C/C++) or low level (e.g., assembly language), and whether interpreted (e.g., Perl), compiled (e.g., C/C++), or Just-In-Time (JIT) compiled from bytecode (e.g., Java). Any known or convenient computer, regardless of architecture, should be capable of executing machine code compiled or otherwise assembled from any language into machine code that is compatible with the computer's architecture.

PIG. 7 depicts a flowchart 700 of an example of a method for message log aggregation and provisioning. In the example of FIG. 7, the flowchart 700 starts at module 702 where a plurality of devices are identified in association with a user profile. Thus, at least one user of a system will have a plurality (i.e., two or more) devices that are capable of generating message logs. This is not intended to mean that all users of a system must have two devices that are capable of generating message logs. Indeed, as explained later with reference to FIG. 8, it is possible to have users associated with a single device in certain embodiments and/or implementations.

In the example of FIG. 7, the flowchart 700 continues to module 704 where a plurality of message logs are created for a respective plurality of the devices and to module 706 where the plurality of message logs are aggregated. Thus, each device has an associated message log. It should be noted that, in a non-limiting embodiment, the message logs could be created simultaneously as a single log, which would have the effect of, essentially, combining modules 704 and 706.

In the example of FIG. 7, the flowchart 700 continues to module 708 where a request for one of the plurality of message logs is received from a respective one of the plurality of devices. It may be noted that the requesting device may have a NULL message log associated with it, and could still request a message log. Thus, any device associated with a user profile could request a message log. It may also be noted that a device could become identified in association with a user profile (702) after having aggregated the plurality of message logs (708), and still request a message log.

In the example of FIG. 7, the flowchart 700 continues to module 710 where at least a portion of the aggregated message log, including a portion of at least a subplurality of the plurality of message logs, is compiled into a format suitable for display on a device associated with the requested message log. Message logs can become quite long over time. While, as used herein, it is intended that, in some cases where a message log is referred to as "sent" or "provided" that only a portion of the message log may or may not be provided. In the example of FIG. 7, a portion is intended to mean the entire portion or a subportion of the aggregated message log. The message logs may be stored, as they are generated, in the aggregated message log. Indeed, the system may or may not, depending upon the implementation, even be able to distinguish between a message log generated by one device and a message generated by another device. The aggregated message log may be compiled into a format that is suitable for display on a requesting device. If the system knows the device is used in association with the user profile, the system may also have formatting preferences. In some cases, the aggregated message log may be stored in a format that does not require changing for some or all of the plurality of devices, in which case module 710 is obviated, at least to the extent compiling is called for.

In the example of FIG. 7, the flowchart 700 continues to module 712 where at least a portion of the aggregated message log is provided in response to the request.

FIG. 8 depicts a flowchart 800 of another example of a method for message log aggregation and provisioning. In the example of FIG. 8, the flowchart 800 starts at module 802 where a first device is identified in association with a user profile. The device may be any device that is capable of sending or receiving messages (typically, though not necessarily, both sending and receiving). Such a device may include, by way of example but not limitation, a browser embodied in a computer-readable medium.

In the example of FIG. 8, the flowchart 800 continues to module 804 where an online platform is provided that sends messages to and from the first device. In a non-limiting embodiment, the online platform would forward outgoing messages from the first device and forward incoming messages from a sender (not shown) to the first device. Thus, the online platform can serve as an intermediary between sender and recipient. Advantageously, the online platform can serve as intermediary for all devices associated with a user, enabling an aggregated message log to be generated from all of the user's devices with messages that pass through the online platform.

In the example of FIG. 8, the flowchart 800 continues to module *806* where an aggregated message log is created from the messages sent to and from the first device. The message log may be aggregated from entries associated with each message sent to or from the first device. Advantageously, the message log can also be aggregated from entries associated with messages sent to or from other devices associated with the user. For example, the message log could aggregate both, e.g., cell phone chat histories and, e.g., game console chat histories into the message log for a user.

In the example of FIG. 8, the flowchart 800 continues to module 808 where a second device is identified in association with the user profile. This module is optional because the user could use a single device (i.e., the first device) to send and receive messages and to receive the aggregated message log.

In the example of FIG. 8, the flowchart 800 continues to module 810 where the aggregated message log is made available from the online platform to the second device. Thus, the first device could include, e.g., a cell phone and the second device could include, e.g., a browser embodied in a computer readable medium on a laptop computer. The user would be able to access the cell phone chatlog online on the laptop computer. As was noted with reference to module 808, the first device and the second device may be the same device. That is, an IM message log associated with, e.g., a smart phone could be aggregated at the online platform, and then the same smart phone could be used to access the aggregated message log.

As used herein, the term "embodiment" means an embodiment that serves to illustrate by way of example but not limitation.

It will be appreciated to those skilled in the art that the preceding examples and embodiments are exemplary and not limiting to the scope of the present invention. It is intended that all permutations, enhancements, equivalents, and improvements thereto that are apparent to those skilled in the art upon a reading of the specification and a study of the drawings are included within the true spirit and scope of the present invention. It is therefore intended that the following appended claims include all such modifications, permutations and equivalents as fall within the true spirit and scope of the present invention.

CLAIMS

1. A system comprising:

a message history database, embodied in a computer-readable medium, including a mobile device message log associated with a user;

a database interface for accessing the message history database;

a history aggregation engine, embodied in a computer-readable medium and coupled to the database interface, that aggregates log entries associated with messages sent to or from a mobile device associated with the user for storage in the message history database;

an online history provisioning engine, embodied in a computer-readable medium and coupled to the database interface, that provides an aggregated .message log associated with the user from the message history database to a requesting device.

2. The system of claim I, wherein the mobile device is one of a plurality of mobile devices and the mobile device message log is one of a plurality of mobile device message logs, further comprising:

.a plurality of mobile devices associated with the user and respectively associated with a plurality of mobile device message logs;

a plurality of networks coupling the plurality of devices to the history aggregation engine;

wherein, in operation, the history aggregation engine receives messages from the plurality of mobile devices through the plurality of networks, and forms the mobile device message logs thereby.

3. The system of claim 1, wherein the mobile device is one of a plurality of mobile devices and the mobile device message log is one of a plurality of mobile device message logs, further comprising:

a plurality of user devices associated with the user and respectively associated with a plurality of message logs;

a plurality of networks coupling the plurality of devices to the history provisioning engine;

wherein, in operation, the history provisioning engine receives a request from a device of the plurality of devices and provides the aggregated message log to the device in a format suitable for display on the device.

4. The system of claim 1, further comprising:

a user profile database, embodied in a computer-readable medium, including a user record that sufficient data to identify a plurality of message log formats associated with a user;

the message history database, including a message log associated with the user;

wherein, in operation, the message log provisioning engine provides the message log in one of the plurality of message log formats.

5. The system of claim 1, wherein, in operation, the history aggregation engine compiles the aggregated message log on the fly.

6. The system of claim 1, wherein the plurality of message logs are stored in the message history database in a respective plurality of message log formats.

7. The system of claim 1, wherein, in operation, the history aggregation engine compiles at least a portion of the aggregated message log in advance of receiving a request from the requesting device.

8. The system of claim 1, wherein the plurality of message logs are stored in the message history database in an aggregated log format.

9. A method comprising:

identifying a plurality of devices in association with a user profile;

creating a plurality of message logs for respective ones of the plurality of devices;

aggregating the plurality of message logs;

receiving a request for one of the plurality of-message logs from a respective one of the plurality of devices;

compiling at least a portion of the aggregated message log, including a portion of at least a subplurality of the plurality of message logs, into a format suitable for display on a device associated with the requested message log;

providing the at least a portion of the aggregated message log in response to the request.

10. The method of claim 9, further comprising making message log entries when a message is sent to or from one of the plurality of devices.

11. The method of claim 9, further comprising:

storing one of the plurality of message logs on a respective one of the plurality of devices;

uploading the one of the plurality of message logs for aggregation.

12. The method of claim 9, wherein the request is from the device associated with the requested message log.

13. The method of claim 9, further comprising compiling the aggregated message log on the fly.

14. The method of claim 9, further comprising compiling at least a portion of the aggregated message log prior to receiving the request.

15. The method of claim 9, further comprising storing the aggregated message log in a plurality of formats, wherein at least one of the plurality of formats is suitable for display on each of the plurality of devices.

16. The method of claim 9, further comprising displaying the aggregated message log on the device associated with the requested message log.

17. A method comprising:

identifying a device in association with a user profile;

providing an online platform that receives messages from and sends messages to the device;

creating an aggregated log from messages sent to and from the device.

18. The method of claim 17, further comprising creating an aggregated log from a group chat message.

19. The method of claim 17, further comprising timestamping message entries in the aggregated log.

20. The method of claim 17, further comprising making the aggregated log available online.

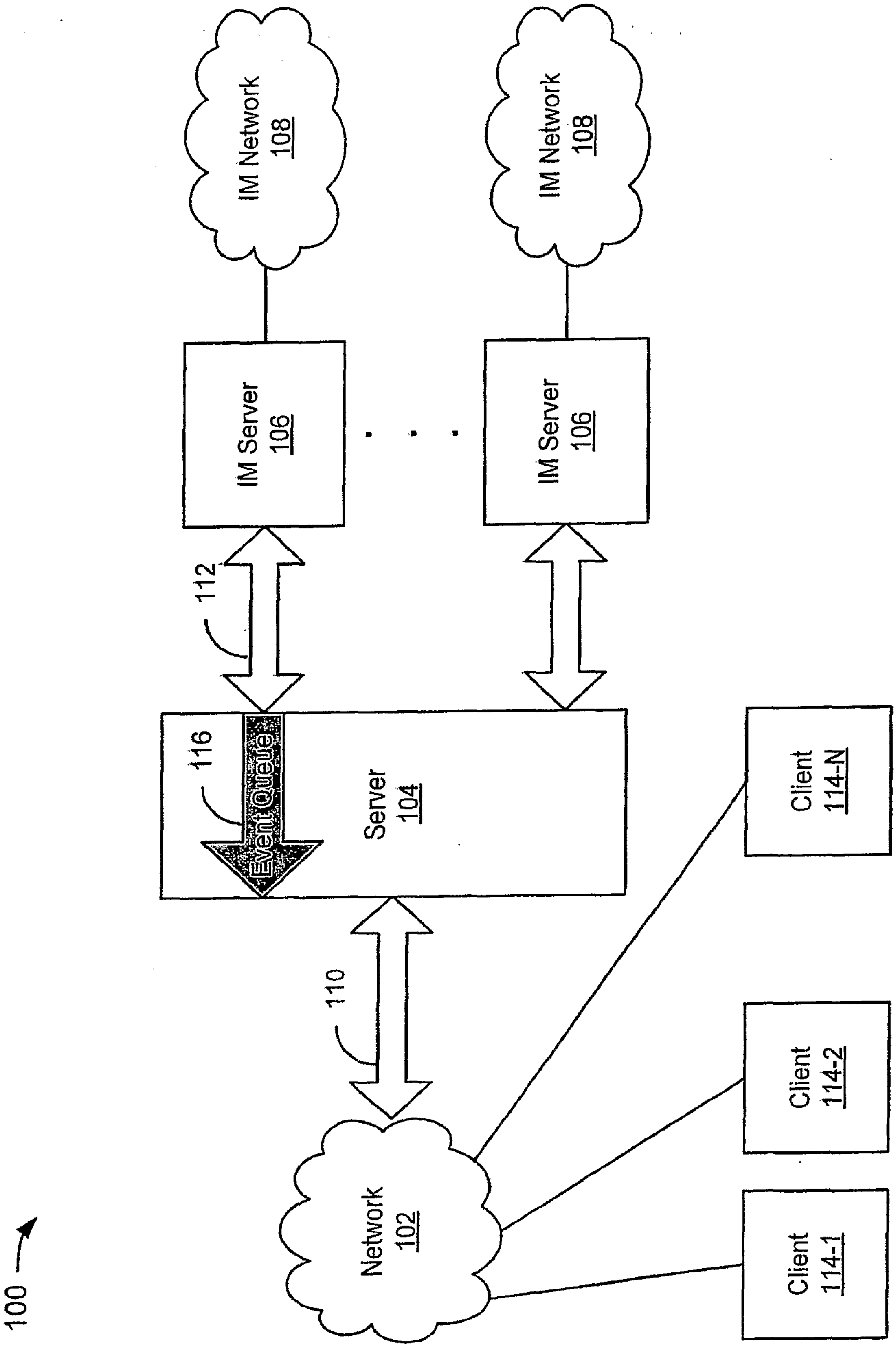
100
IM Network 108
IM Network 108
IM Server 106
IM Server 106
112
116
Event Queue
Server 104
110
Network 102
Client 114-1
Client 114-2
Client 114-N

FIG. 1

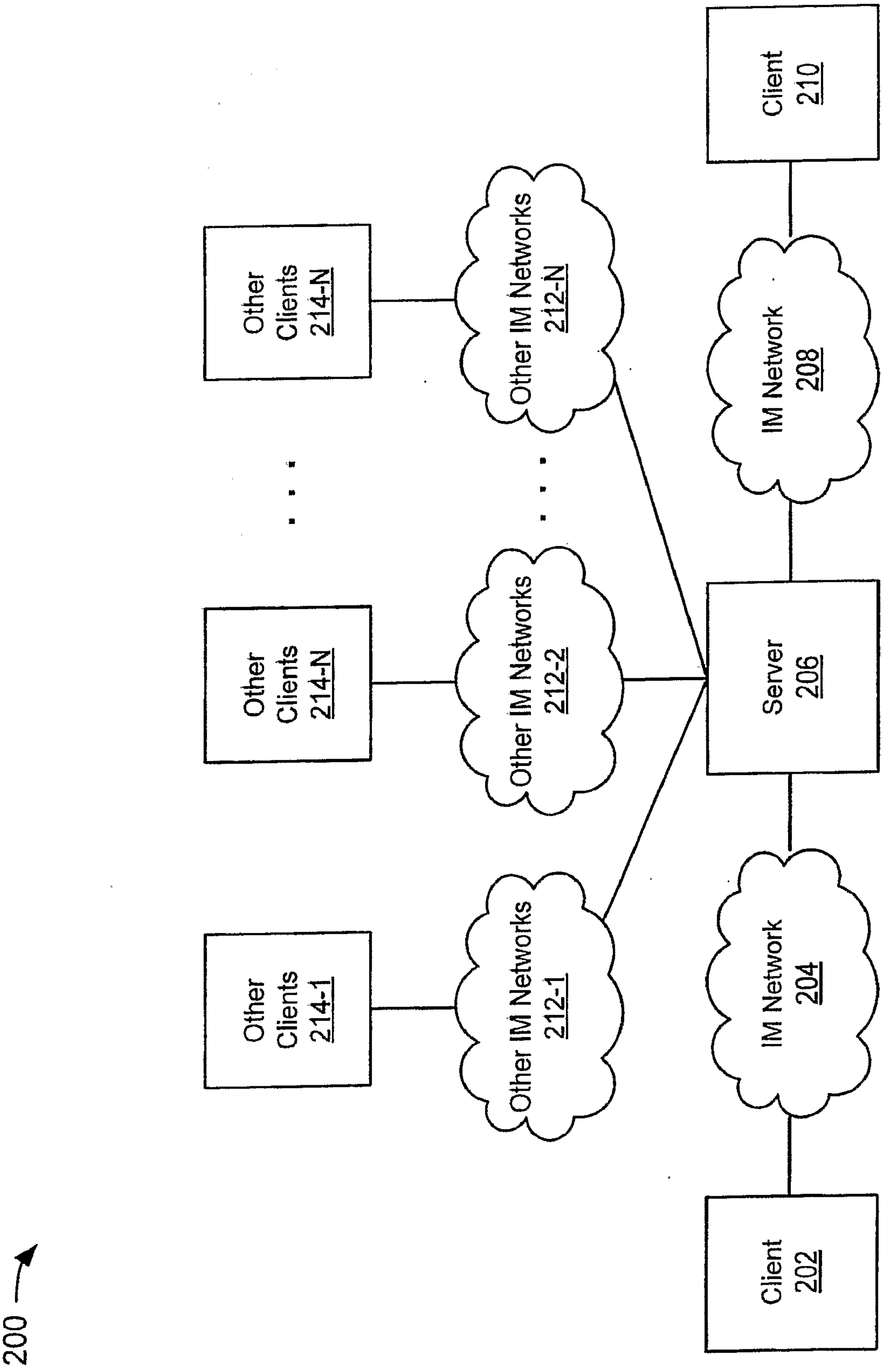
200
Client 202
IM Network 204
Server 206
IM Network 208
Client 210
Other IM Networks 212-1
Other IM Networks 212-2
Other IM Networks 212-N
Other Clients 214-1
Other Clients 214-N
Other Clients 214-N

FIG. 2

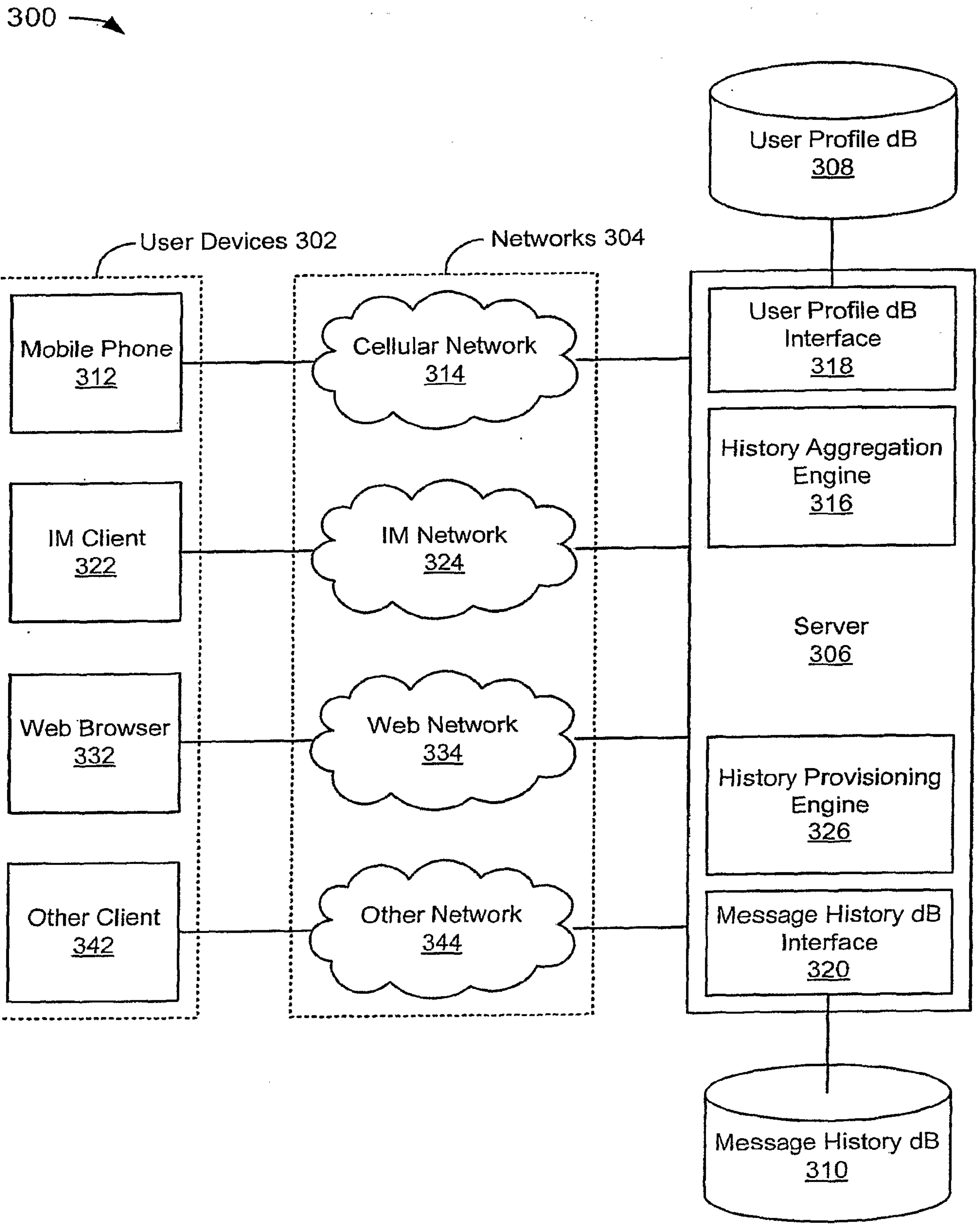
300
User Profile dB
308
User Devices 302
Networks 304
User Profile dB
Interface
318
Mobile Phone
312
Cellular Network
314
History Aggregation
Engine
316
IM Client
322
IM Network
324
Server
306
Web Browser
332
Web Network
334
History Provisioning
Engine
326
Other Client
342
Other Network
344
Message History dB
Interface
320
Message History dB
310

FIG. 3

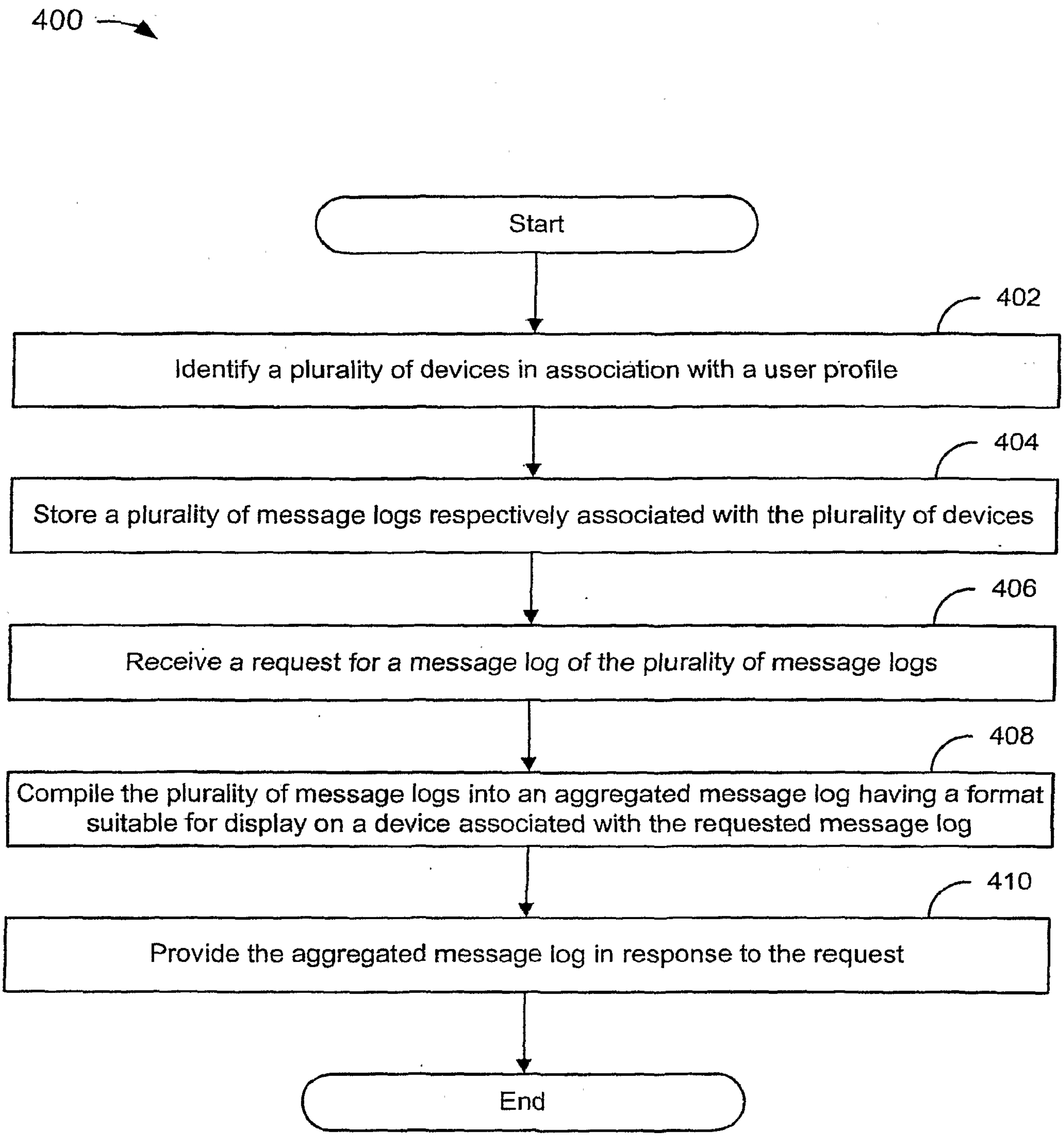
400
Start
402
Identify a plurality of devices in association with a user profile
404
Store a plurality of message logs respectively associated with the plurality of devices
406
Receive a request for a message log of the plurality of message logs
408
Compile the plurality of message logs into an aggregated message log having a format suitable for display on a device associated with the requested message log
410
Provide the aggregated message log in response to the request
End

FIG. 4

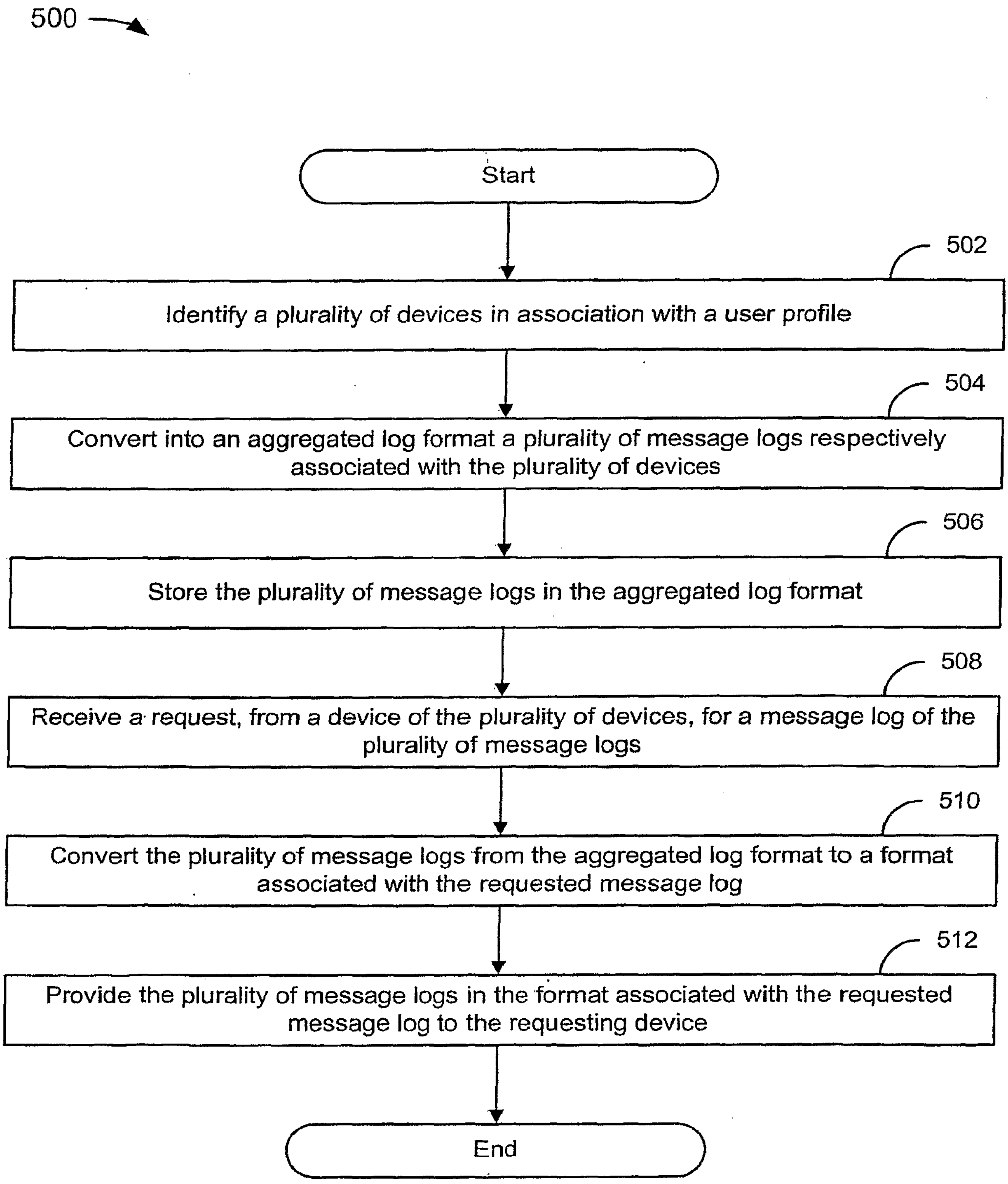
500
Start
502
Identify a plurality of devices in association with a user profile
504
Convert into an aggregated log format a plurality of message logs respectively associated with the plurality of devices
506
Store the plurality of message logs in the aggregated log format
508
Receive a request, from a device of the plurality of devices, for a message log of the plurality of message logs
510
Convert the plurality of message logs from the aggregated log format to a format associated with the requested message log
512
Provide the plurality of message logs in the format associated with the requested message log to the requesting device
End

FIG. 5

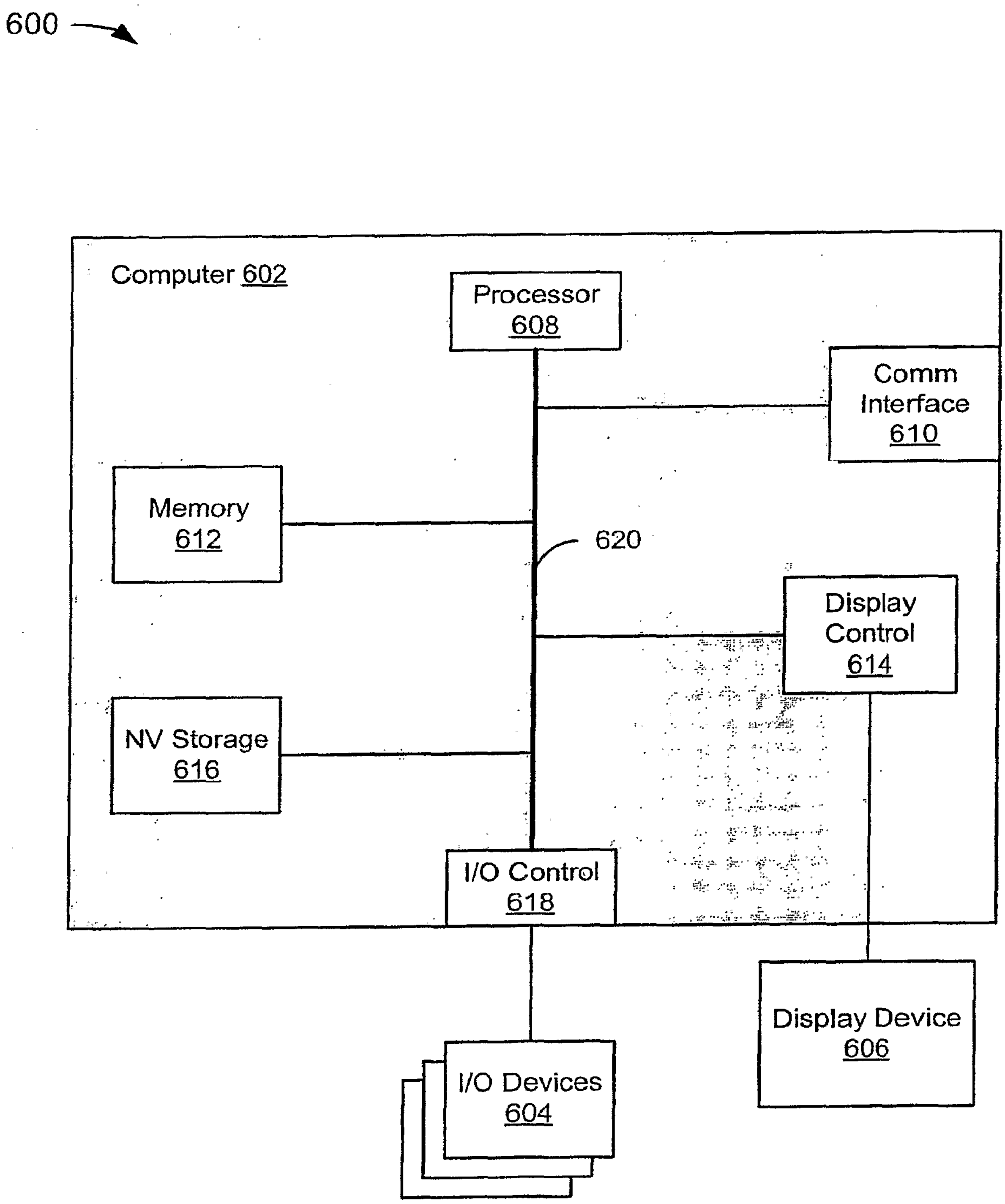
600
Computer 602
Processor 608
Comm Interface 610
Memory 612
620
Display Control 614
NV Storage 616
I/O Control 618
I/O Devices 604
Display Device 606

FIG. 6

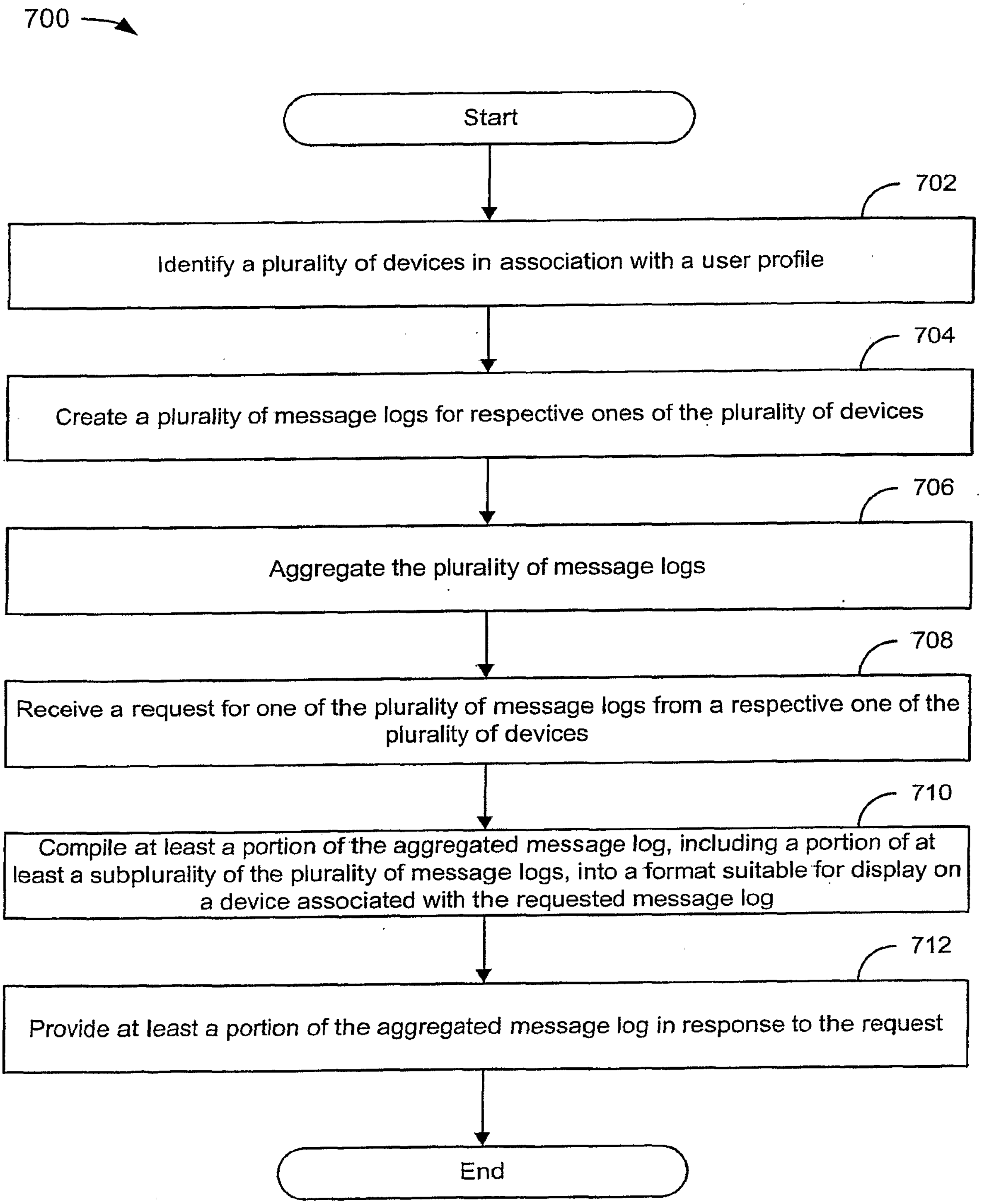
700
Start
702
Identify a plurality of devices in association with a user profile
704
Create a plurality of message logs for respective ones of the plurality of devices
706
Aggregate the plurality of message logs
708
Receive a request for one of the plurality of message logs from a respective one of the plurality of devices
710
Compile at least a portion of the aggregated message log, including a portion of at least a subplurality of the plurality of message logs, into a format suitable for display on a device associated with the requested message log
712
Provide at least a portion of the aggregated message log in response to the request
End

FIG. 7

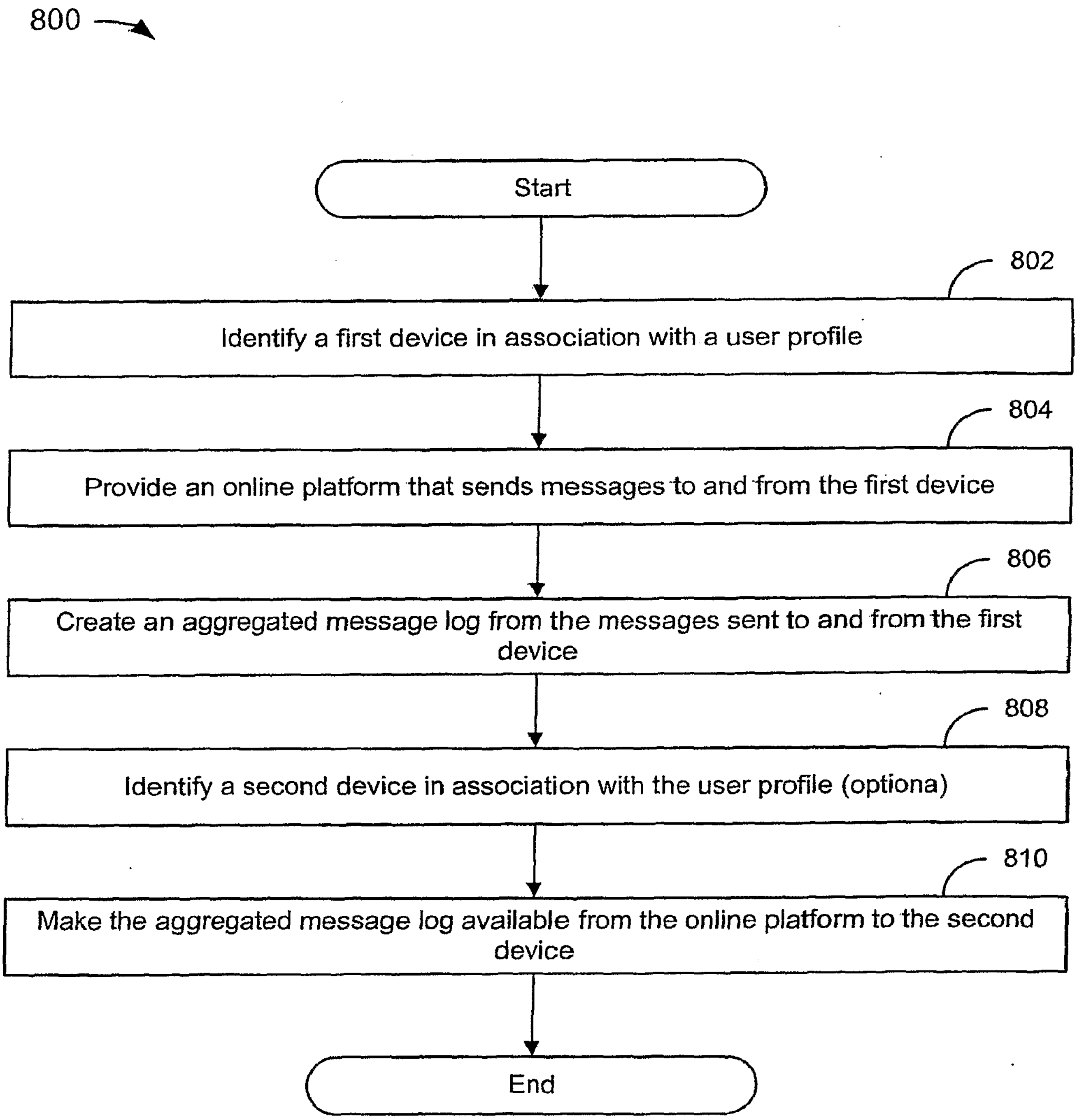
800
Start
802
Identify a first device in association with a user profile
804
Provide an online platform that sends messages to and from the first device
806
Create an aggregated message log from the messages sent to and from the first device
808
Identify a second device in association with the user profile (optiona)
810
Make the aggregated message log available from the online platform to the second device
End

FIG. 8