



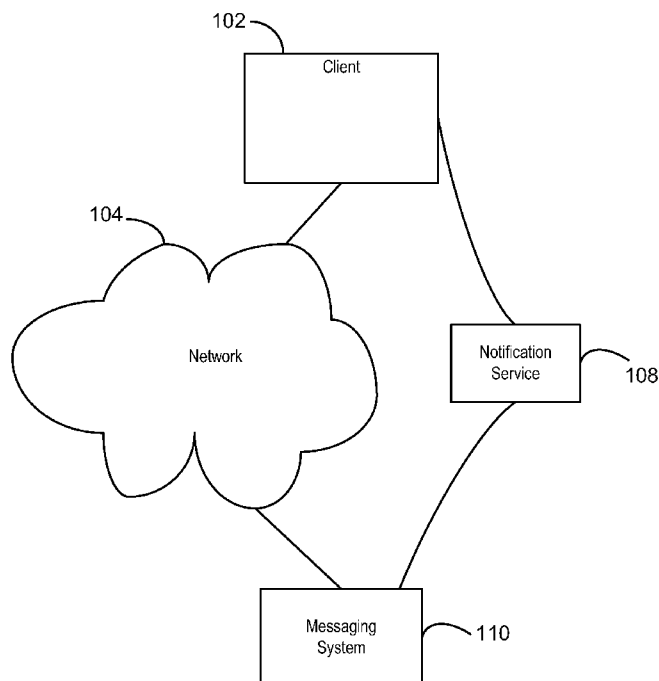
- (51) International Patent Classification:
G06Q 50/30 (2012.01)
- (21) International Application Number:
PCT/US2012/027928
- (22) International Filing Date:
7 March 2012 (07.03.2012)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
13/049,913 17 March 2011 (17.03.2011) US
- (71) Applicant (for all designated States except US): **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, WA 98052-6399 (US).
- (72) Inventors: **FIRSTENBERG, Yosef**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US). **TANG, Lan**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 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, 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, 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, MD, 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).

[Continued on next page]

(54) Title: MESSAGING FOR NOTIFICATION-BASED CLIENTS



(57) Abstract: The claimed subject matter provides a method for communicating with a client. The method includes receiving a notification token that identifies the client. The method further includes registering the client to receive a message when the client is not connected to the server. The method also includes sending the message to the client when the client does not have a connection to the server, based on the registering and the notification token. The message may be sent using a notification channel.

FIG. 1



**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:

- *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

MESSAGING FOR NOTIFICATION-BASED CLIENTS

BACKGROUND

[0001] Client-server communications systems may vary in terms of when the server may send a communication to the client. For example, communications between a website
5 (server) and a client may follow a request-response model. In this model, the client may be reachable from the server if the client is sending a request to the server. The client may otherwise be unreachable. As such, the server may only send communications to the client in response to a client request. In this model, the client request may result in an open network connection over which the server may send communications.

10 [0002] This model differs from traditional messaging systems. Traditional messaging systems assumed the existence of a connection between clients and servers. Because the connection persists, the server may send a communication to the client at any time.

[0003] On increasing numbers of platforms, e.g., mobile devices, maintaining network connections is resource intensive. For example, mobile devices may quickly drain battery
15 power by trying to maintain a network connection. As such, these platforms may not maintain network connections when client applications are not active.

[0004] Maintaining a network connection may be further impeded by limitations of the networks themselves. In some cases, networks may not be reliable. For example, cell phones lose their network connections in regions without appropriate cellular towers.

20 [0005] Typical approaches to these types of connectivity issues still rely on creating open network connections between the client and the server. Depending on open network connections for communications between clients and servers increases the demand on the limited resources of increasing numbers of client devices.

SUMMARY

25 [0006] The following presents a simplified summary of the innovation in order to provide a basic understanding of some aspects described herein. This summary is not an extensive overview of the claimed subject matter. It is intended to neither identify key or critical elements of the claimed subject matter nor delineate the scope of the subject innovation. Its sole purpose is to present some concepts of the claimed subject matter in a simplified form as a prelude to the more detailed description that is presented later.
30

[0007] The subject innovation relates to a method and a system for communicating with a client that is not connected to a server. The client may register notification channel information with the server. The server may send a message to the client using the

notification channel. The messages may be included in a notification message to the client. Alternatively, the message may be sent using a shoulder tapping technique.

[0008] In one embodiment, the client may subscribe to buddies, services, or networks. By subscribing, the client may be automatically notified of corresponding updates over the notification channels. The following description and the annexed drawings set forth in detail certain illustrative aspects of the claimed subject matter. These aspects are indicative, however, of a few of the various ways in which the principles of the innovation may be employed and the claimed subject matter is intended to include all such aspects and their equivalents. Other advantages and novel features of the claimed subject matter will become apparent from the following detailed description of the innovation when considered in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. 1 is a block diagram of a system in accordance with the claimed subject matter;

[0010] FIG. 2 is a process flow diagram of a communications method for clients in accordance with the claimed subject matter;

[0011] FIG. 3 is a message flow diagram for creating a client entity in the network in accordance with the claimed subject matter;

[0012] FIG. 4 is a message flow diagram for sending a message in accordance with the claimed subject matter;

[0013] FIG. 5 is a message flow diagram for subscribing in accordance with the claimed subject matter;

[0014] FIG. 6 is a message flow diagram for querying a buddy presence in accordance with the claimed subject matter;

[0015] FIG. 7 is a message flow diagram for deleting a client entity in accordance with the claimed subject matter;

[0016] FIG. 8 is a message flow diagram for multiparty conversation in accordance with the claimed subject matter;

[0017] FIG. 9 is a block diagram of an exemplary networking environment wherein aspects of the claimed subject matter can be employed; and

[0018] FIG. 10 is a block diagram of an exemplary operating environment for implementing various aspects of the claimed subject matter.

DETAILED DESCRIPTION

[0019] The claimed subject matter is described with reference to the drawings, wherein like reference numerals are 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 subject innovation. It may be evident, however, that the claimed subject matter may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate describing the subject innovation.

[0020] As utilized herein, terms “component,” “system,” “client” and the like are intended to refer to a computer-related entity, either hardware, software (e.g., in execution), and/or firmware, or a combination thereof. For example, a component can be a process running on a processor, an object, an executable, a program, a function, a library, a subroutine, and/or a computer or a combination of software and hardware.

[0021] By way of illustration, both an application running on a server and the server can be a component. One or more components can reside within a process and a component can be localized on one computer and/or distributed between two or more computers. The term “processor” is generally understood to refer to a hardware component, such as a processing unit of a computer system.

[0022] 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” as used herein is intended to encompass a computer program accessible from any non-transitory computer-readable device, or media.

[0023] Non-transitory computer-readable storage media can include but are not limited to magnetic storage devices (e.g., hard disk, floppy disk, and magnetic strips, among others), optical disks (e.g., compact disk (CD), and digital versatile disk (DVD), among others), smart cards, and flash memory devices (e.g., card, stick, and key drive, among others). In contrast, computer-readable media generally (i.e., not necessarily storage media) may additionally include communication media such as transmission media for wireless signals and the like.

[0024] 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. 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 preferred or advantageous over other aspects or designs.

5 [0025] For increasing numbers of mobile platforms, messaging systems may use an open network connection between the client and the server to enable the server to send messages to the client. However, when the client is not able to establish a long-lived connection, the connection may be simulated.

10 [0026] As stated previously, mobile platforms introduce challenges with respect to maintaining the connection between the client and the server, including: 1) due to the nature of the communication channel, e.g., wireless, the connection may not be reliable, 2) maintaining the connection to the server consumes battery resources which are limited on those platforms, and 3) wireless connections may be more limited in terms of bandwidth than wired connections.

15 [0027] To address these challenges, messaging systems may reduce bandwidth consumption, and add resiliency logic to the transport layer of a communications protocol. Adding resiliency logic to the transport layer typically enables messaging systems to recover from intermediate connectivity issues.

20 [0028] FIG. 1 is a block diagram of a system 100 in accordance with the claimed subject matter. The system 100 includes a client 102, a network 104, a notification service 108, and a messaging system 110. The client 102 may be a device with a client application that communicates over the network 104 with the messaging system 110. The client 102 may be any type of communications or computing device, such as a mobile telephone, personal digital assistant, a desktop computer, etc. The platform on which the client 102 runs may include a set of APIs to interact with a notification service 108. The notification service 108 may be another cloud service provided by a platform that the client 102 runs on. For example, on a Windows Phone, the notification service 108 may be the Microsoft Push Notification service (MPN). An iPhone client may use the Apple Push Notification service (APN). For clients 102 that run on a platform that does not provide native notification service APIs, the messaging system 110 may provide an alternative.

30 [0029] The notification service 108 may provide one or more notification channels that a cloud service may use to send information to the client 102 that a) may conserve battery life and b) may be configured to overcome inherent network challenges described above, such as intermittent connectivity to the network 104. Usually the connection from

the client 102 to the notification service 108 may be allowed to be opened to preserve client resources.

{0030} The network 104 may be any network or collection of networks, capable of supporting communications between the client 102 and the messaging system 110. The network 104 may be the Internet, a wide area network, a local area network, etc. The Internet is also referred to herein as a cloud.

{0031} The notification service 108 may be used as part of a messaging system 110, used for communicating messages from the messaging system 110 to the client 102. As such, the messaging system 110 may enable typical messaging system functionality, including: messaging, signaling, presence, audio, video, etc. In one embodiment, this functionality may be provided in the absence of a connection between the client 102 and the messaging system 110.

{0032} Presence may represent the availability and capabilities of the client 102 at a given moment for certain operations. In typical messaging systems, the availability of the client 102 is closely connected with a connection state. Where is no connection, the client 102 is assumed to be unavailable. In embodiments of the invention, when there is no connection, the client 102 may be available through the notification service 108.

{0033} Instead of communicating through an open network connection, the messaging system 110 may communicate with the client 102 using the notification service 108. The messaging system 110 may use a protocol that natively supports the notification service 108 as part of a client connectivity scheme. In one embodiment, the protocol may support this type of client 102, i.e., a notification-based client, as a first class entity, and provide constructs that natively fit various platforms for the client 102.

{0034} In such an embodiment, the messaging system 110 may use a non-connected client mode. In this mode, a client entity may be provided by a cloud service, without necessarily having an open network session. The client 102 may be reachable using other means, such as the notification service 108.

{0035} The protocol may include commands, for registering and unregistering clients, updating clients, sending messages, subscribing to buddy lists, querying for buddies, querying and delivering messages, and supporting multiparty communications. These commands are described in greater detail with reference to FIGs. 4-9.

{0036} Some differences of this protocol from existing protocols include: 1) the removal of the session as a condition of the existence of the client entity in the cloud, and 2) the use of the non-connected mode as a way to interact with the cloud service.

[0037] The cloud service may store data about the notification service 108. The notification service 108, or an open network session (if it exists), may be used to communicate back to the client 102. This protocol may natively support a variety of notification platforms. Using this protocol, the messaging system 110 may also provide notification services to clients that do not provide notification services, e.g., web clients. In one embodiment, this may be done using the same semantics as defined by other notification services over a persistent transport layer service.

[0038] In one embodiment, the messaging system 110 may generate an authentication token as a way to authenticate the client and to control the lifetime of the client entity in the cloud. The messaging system 110 may store the notification channel information to be used to communicate back to the client 102. The notification channel is a way for the messaging system 110 to send a message to the client 102 using the notification service 108. Additionally, the messaging system 110 may store information about any potential open network session, if such a connection exists. This connection information may be used as an alternative way to communicate back to the client 102. The client 102 may send the authentication token with every request, expecting to receive authentication failure responses, and be able to re-authenticate to the messaging system 110.

[0039] When the client 102 is sending messages to the messaging system 110, the client may use connected or non-connected means. In one embodiment, the client 102 may use a non-connected channel. In an alternative non-connected means, the client 102 may open a temporary channel for ad-hoc communications and use TCP-based commands. This may be done without explicitly connecting or re-connecting to the messaging system 110.

[0040] In terms of connected means, the client 102 may explicitly connect to the messaging system 110 for the duration of one or more conversations. The client 102 may be limited to a single connection to the cloud at any given time, which may be enforced by the messaging system 110.

[0041] The protocol may not define the client presence status in the cloud, but rather the mechanism in which the client 102 may update its specific presence information, e.g., availability and capabilities. The protocol may also provide a way for those with a relationship to the user (for example, a connection on a social or professional website) to receive updates when applicable. Because of connectivity and process privilege constraints, the client 102 may not accurately update its presence. Instead, the protocol may assume that business logic on the messaging system 110 updates the client presence.

This can be done based on time interval (i.e. “minute from last activity”), notification channel status (i.e. “periodic ping test” or “error responses”), or explicit update by the client 102 itself when possible.

[0042] The protocol may support one way to register or unregister the client entity.

5 However, the messaging system 110 may not enforce registering or unregistering the client 102 using the protocol. There may be other ways to add client entities to the messaging system 110. For example, the messaging system 110 may query an external authoritative source, such as another cloud service that keeps track of the clients 102. As long as the client 102 presents a valid token, the messaging system 110 may serve the
10 requests. The security challenges implicated by this approach may be addressed by using a secured communication channel.

[0043] To further preserve battery life and reduce bandwidth consumption, the protocol may free the client 102 from uploading any data that already exists in any authoritative entity in the cloud. Instead, the client 102 may merely upload client-specific
15 and transient data. The messaging system 110 may store a limited amount of state information for each client. This state may include, for example, notification channel information, access control lists (ACLs), availability and capabilities, subscriptions, cached instant messages (IMs) or notifications, and last activity time information.

[0044] FIG. 2 is a process flow diagram of a communications method 200 for clients
20 102 in accordance with the claimed subject matter. In one embodiment, the messaging system may perform the method 200.

[0045] The method 200 begins at block 202, where the notification service 108 may generate a notification token. The notification token may be used by the messaging system 110 to identify the recipient of the notifications it sends to through the notification
25 service 108. The client 102 may upload the notification token to the messaging system 110 in addition to the notification service information.

[0046] At block 204, the messaging system 110 may register the client 102 to receive communications from the messaging system 110. Registering the client 102 may include storing notification service and token information for the client 102. The notification
30 channel information together with the notification token may provide an address for contacting the client 102 via the notification service 108.

[0047] If, however, the client 102 is not connected to messaging system, and there is a message from the server for the client 102, at block 206, the messaging system 110 may send the message to the non-connected client 102 using the notification service.

[0048] FIGS. 3-8 are message flow diagrams for typical communications scenarios according to the protocol described above in accordance with the claimed subject matter. These commands may be performed in an implementation of the protocol described above. In these figures, the abbreviation EP, shorthand for endpoint, is a reference to the client

5 102.

[0049] The commands of this protocol may be supported by standard HTTP methods. For example, the scenarios illustrated with reference to FIGS. 3-8, show an example implementation of the commands using standard HTTP methods. Standard HTTP methods may include: OPTIONS, GET, HEAD, PUT, POST, DELETE, TRACE, and

10 CONNECT.

[0050] As stated previously, the protocol may include a set of commands for binding, subscribing, sending messages, query and updating presence information, establishing 2 party and multi-part conversations etc.

[0051] The registration command may create a client entity in the messaging system

15 110 and return an authentication token to the client. The un-register command may delete the client entity from the messaging system 110. In one embodiment, the client 102 may specify other client entities to be deleted according to specified permissions.

[0052] Once the client entity is registered to the messaging system 110, the corresponding client 102 may publish its presence using the update client command.

20 Parameters specified in the command may include notification channel information for each operation that the client 102 supports. Parameters may also include a notification type that specifies how notifications are delivered to the client 102. One type of notification may specify that the command is included in the notification payload. Another type may specify that the command is not included, e.g., shoulder tapping.

25 [0053] One embodiment may include a command for sending a message to a buddy without establishing a connection to the cloud. The notification channel or a command to messaging system 110 to get the queued message from the messaging system 110 may be used.

[0054] A command for subscribing may be used to enable the client to subscribe to

30 buddies. Subscribing to buddies enables the client 102 to be automatically notified of updates to the buddies presence statuses. The client 102 may only specify a filter to be applied to the buddy list to subscribe to a subset of the buddies. The messaging system 110 may retrieve contact lists from another cloud service that maintains

the client's buddy list. The client 102 may use this command to subscribe to a specific buddy. The client 102 may similarly unsubscribe to buddies, services, etc.

[0055] The query buddy presence command may enable the client 102 to determine the presence status of one or more contacts of the user. A buddy list may be used to determine which contact's presence status to receive. For example, an empty contact list may result in the messaging system 110 fetching the presence status all the user's contacts. In one embodiment, filter and view selections may be used to improving the accuracy of response, reduce packet size of responses. For example, view options may include various presence elements such as availability for messaging, availability for video calls, etc. Filter options may be used to filter out hidden contacts, groups, offline contacts, etc.

[0056] The message command may deliver messages to the client. In one embodiment, the typical send message commands may be configured to support the notification service 108 and reduce bandwidth consumption. For example, the client 102 may receive these commands in the notification payload or in the body of the response to the get queued messages command. This may be based on client preferences.

[0057] The get queued messages command may be used to retrieve queued notifications and messages. In one embodiment, the client 102 may send poll the messaging system 110 until all cached notifications are retrieved. The protocol may provide a command for getting the queued messages and a separate command for deleting those messages from the messaging system 110.

[0058] A set of commands for creating a temporary multiparty conversation, inviting buddies, sending messages, updating and getting presence information in the context of this multiparty conversation may be used to allow typical multiparty conversation operations.

[0059] FIG. 3 is a message flow diagram 300 for creating a client entity in the messaging system 110 in accordance with the claimed subject matter. The message flow diagram 300 includes messages 304, 306. The messages 304 may authorize client access to the messaging system 110. In response to a request from the client 102, the messaging system 110 may retrieve an access list from a cloud service that maintains the client buddy access control lists 302, and may generate a client authentication token, which is sent to the client 102.

[0060] The messages 306 may register the client 102 with the messaging system 110 for notifications in the non-connected mode. A registration request including the

notification token may be sent to the messaging system 110. This results in the notification token and client presence data being stored with the messaging system 110. The notification token provides an address that the messaging system 110 can use to send messages from the messaging system 110 to the client 102 when not connected. In FIG. 3, the Authentication token is not included in the registration messages 306. In one embodiment, the authentication token may be included.

[0061] The messages described with reference to Figs. 3-8 may be communicated along typical communication channels. The type of communication channel for any particular message may vary depending on the sender and receiver of the message.

[0062] In one embodiment, the client 102 may send a registration command with an authorization token, e.g., a user ticket, to the messaging system 110. Alternatively, the endpoint can send a batch of registration and other commands to minimize the calls from the client to the messaging system 110. The client 102 may not maintain a connection to the messaging system 110 during the process of registration, authentication and setting initial presence information.

[0063] The messaging system 110 may authenticate the client using the authorization token. The messaging system 110 may create a client entity, bind a user to the client, and send back the client authentication token. The client 102 may use the authentication token to identify itself to the messaging system 110 for future communications.

[0064] Once the client entity is created, the client 102 may publish its presence (availability, capabilities, etc.) to the messaging system 110. Alternatively, the messaging system 110 may maintain the client's presence.

[0065] By registering with the messaging system 110, the client 102 may receive notifications on presence changes and messages. The client 102 may also specify whether notifications do not include the actual packet in the notification payload, i.e., shoulder tapping.

[0066] FIG. 4 is a message flow diagram 400 for sending a message in accordance with the claimed subject matter. The message flow diagram 400 includes messages 404, 406. The messages 404 may, while not connected, allow the client 102 to send a message using the authentication token. The messages 406 may enable the client 102 to receive instant messages from other clients 102. As shown, a get queued messages command may be used to retrieve messages from the queue 206.

[0067] FIG. 5 is a message flow diagram 500 for subscribing in accordance with the claimed subject matter. The client 102 may subscribe to a contact's presence by sending a

subscribe command with lists specified in the command's body. The messaging system 110 may retrieve buddy lists from another cloud service that maintains the client's buddy list 502.

[0068] Presence notifications may be sent to the client's messaging system 110. Once
5 the messaging system 110 gets a buddy's presence notification, the messaging system 110 may send a notification to the notification service 108 and identifying the target client using the notification token.

[0069] In one embodiment, the messaging system 110 may cache the notification, and just send a shoulder tap notification to the notification service 108. In such an
10 embodiment, the client 102 may retrieve the cached notification by sending a get queued messages command to the messaging system 110.

[0070] FIG. 6 is a message flow diagram 600 for querying a buddy presence in accordance with the claimed subject matter. As shown, the client 102 may query the presence of a buddy using the authentication token.

15 [0071] FIG. 7 is a message flow diagram 700 for unregistering a client entity in accordance with the claimed subject matter. This request may specify the authentication token in a unregister command to the messaging system 110. The messaging system 110 may delete the client entity represented by the client authentication token from the cloud.

[0072] To unregister the client 102 from the messaging system 110, the messaging
20 system 110 may set a client state to unavailable. As such, any further messages to the client 102 may be dropped, with an error returned to the sender.

[0073] FIG. 8 is a message flow diagram 800 for multiparty conversation in accordance with the claimed subject matter. For multiparty conversation, a roster list may be hosted on a host 802. The host 802 may be a component of the messaging system 110.
25 The conversation may be identified as a temporary group.

[0074] As stated previously, existing messaging systems may be modified to accommodate the non-connected mode described above. These modifications may introduce issues with load balancing, throttling, and geo-hosting. In one embodiment, the messaging system 110 may use various load-balancing techniques to address the change in
30 the client connection mode. A global load-balancing system may be used for balancing the load across multiple geographical cloud locations.

[0075] Additionally, the messaging system 110 may use various throttling techniques to control the use of its resources by the clients and to protect itself from malicious usage.

[0076] FIG. 9 is a block diagram of an exemplary networking environment 900 wherein aspects of the claimed subject matter can be employed. Moreover, the exemplary networking environment 900 may be used to implement a system and method of communicating with non-connected clients.

5 [0077] The networking environment 900 includes one or more client(s) 910. The client(s) 910 can be hardware and/or software (e.g., threads, processes, computing devices). As an example, the client(s) 910 may be computers providing access to servers over a communication framework 940, such as the Internet.

[0078] The environment 900 also includes one or more server(s) 920. The server(s)
10 920 can be hardware and/or software (e.g., threads, processes, computing devices). The server(s) 920 may include network storage systems. The server(s) may be accessed by the client(s) 910.

[0079] One possible communication between a client 910 and a server 920 can be in the form of a data packet adapted to be transmitted between two or more computer
15 processes. The environment 900 includes a communication framework 940 that can be employed to facilitate communications between the client(s) 910 and the server(s) 920.

[0080] The client(s) 910 are operably connected to one or more client data store(s) 950 that can be employed to store information local to the client(s) 910. The client data store(s) 950 may be located in the client(s) 910, or remotely, such as in a cloud server.
20 Similarly, the server(s) 920 are operably connected to one or more server data store(s) 930 that can be employed to store information local to the servers 920.

[0081] With reference to FIG. 10, an exemplary operating environment 1000 is shown for implementing various aspects of the claimed subject matter. The exemplary operating environment 1000 includes a computer 1012. The computer 1012 includes a processing
25 unit 1014, a system memory 1016, and a system bus 1018.

[0082] The system bus 1018 couples system components including, but not limited to, the system memory 1016 to the processing unit 1014. The processing unit 1014 can be any of various available processors. Dual microprocessors and other multiprocessor architectures also can be employed as the processing unit 1014.

30 [0083] The system bus 1018 can be any of several types of bus structure(s) including the memory bus or memory controller, a peripheral bus or external bus, and/or a local bus using any variety of available bus architectures known to those of ordinary skill in the art. The system memory 1016 comprises non-transitory computer-readable storage media that includes volatile memory 1020 and nonvolatile memory 1022.

[0084] The basic input/output system (BIOS), containing the basic routines to transfer information between elements within the computer 1012, such as during start-up, is stored in nonvolatile memory 1022. By way of illustration, and not limitation, nonvolatile memory 1022 can include read only memory (ROM), programmable ROM (PROM),
5 electrically programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), or flash memory.

[0085] Volatile memory 1020 includes random access memory (RAM), which acts as external cache memory. By way of illustration and not limitation, RAM is available in many forms such as static RAM (SRAM), dynamic RAM (DRAM), synchronous DRAM
10 (SDRAM), double data rate SDRAM (DDR SDRAM), enhanced SDRAM (ESDRAM), SynchLink™ DRAM (SLDRAM), Rambus® direct RAM (RDRAM), direct Rambus® dynamic RAM (DRDRAM), and Rambus® dynamic RAM (RDRAM).

[0086] The computer 1012 also includes other non-transitory computer-readable media, such as removable/non-removable, volatile/non-volatile computer storage media.

15 FIG. 10 shows, for example a disk storage 1024. Disk storage 1024 includes, but is not limited to, devices like a magnetic disk drive, floppy disk drive, tape drive, Jaz drive, Zip drive, LS-100 drive, flash memory card, or memory stick.

[0087] In addition, disk storage 1024 can include storage media separately or in combination with other storage media including, but not limited to, an optical disk drive
20 such as a compact disk ROM device (CD-ROM), CD recordable drive (CD-R Drive), CD rewritable drive (CD-RW Drive) or a digital versatile disk ROM drive (DVD-ROM). To facilitate connection of the disk storage devices 1024 to the system bus 1018, a removable or non-removable interface is typically used such as interface 1026.

[0088] It is to be appreciated that FIG. 10 describes software that acts as an
25 intermediary between users and the basic computer resources described in the suitable operating environment 1000. Such software includes an operating system 1028. Operating system 1028, which can be stored on disk storage 1024, acts to control and allocate resources of the computer system 1012.

[0089] System applications 1030 take advantage of the management of resources by
30 operating system 1028 through program modules 1032 and program data 1034 stored either in system memory 1016 or on disk storage 1024. It is to be appreciated that the claimed subject matter can be implemented with various operating systems or combinations of operating systems.

[0090] A user enters commands or information into the computer 1012 through input device(s) 1036. Input devices 1036 include, but are not limited to, a pointing device (such as a mouse, trackball, stylus, or the like), a keyboard, a microphone, a joystick, a satellite dish, a scanner, a TV tuner card, a digital camera, a digital video camera, a web camera, and/or the like. The input devices 1036 connect to the processing unit 1014 through the system bus 1018 via interface port(s) 1038. Interface port(s) 1038 include, for example, a serial port, a parallel port, a game port, and a universal serial bus (USB).

[0091] Output device(s) 1040 use some of the same type of ports as input device(s) 1036. Thus, for example, a USB port may be used to provide input to the computer 1012, and to output information from computer 1012 to an output device 1040.

[0092] Output adapter 1042 is provided to illustrate that there are some output devices 1040 like monitors, speakers, and printers, among other output devices 1040, which are accessible via adapters. The output adapters 1042 include, by way of illustration and not limitation, video and sound cards that provide a means of connection between the output device 1040 and the system bus 1018. It can be noted that other devices and/or systems of devices provide both input and output capabilities such as remote computer(s) 1044.

[0093] The computer 1012 can be a server hosting various software applications in a networked environment using logical connections to one or more remote computers, such as remote computer(s) 1044. The remote computer(s) 1044 may be client systems configured with web browsers, PC applications, mobile phone applications, and the like.

[0094] The remote computer(s) 1044 can be a personal computer, a server, a router, a network PC, a workstation, a microprocessor based appliance, a mobile phone, a peer device or other common network node and the like, and typically includes many or all of the elements described relative to the computer 1012.

[0095] For purposes of brevity, only a memory storage device 1046 is illustrated with remote computer(s) 1044. Remote computer(s) 1044 is logically connected to the computer 1012 through a network interface 1048 and then physically connected via a communication connection 1050.

[0096] Network interface 1048 encompasses wire and/or wireless communication networks such as local-area networks (LAN) and wide-area networks (WAN). LAN technologies include Fiber Distributed Data Interface (FDDI), Copper Distributed Data Interface (CDDI), Ethernet, Token Ring and the like. WAN technologies include, but are not limited to, point-to-point links, circuit switching networks like Integrated Services

Digital Networks (ISDN) and variations thereon, packet switching networks, and Digital Subscriber Lines (DSL).

[0097] Communication connection(s) 1050 refers to the hardware/software employed to connect the network interface 1048 to the bus 1018. While communication connection 1050 is shown for illustrative clarity inside computer 1012, it can also be external to the computer 1012. The hardware/software for connection to the network interface 1048 may include, for exemplary purposes only, internal and external technologies such as, mobile phone switches, modems including regular telephone grade modems, cable modems and DSL modems, ISDN adapters, and Ethernet cards.

[0098] An exemplary processing unit 1014 for the server may be a computing cluster comprising Intel® Xeon CPUs. The disk storage 1024 may comprise an enterprise data storage system, for example, holding thousands of impressions.

[0099] What has been described above includes examples of the subject innovation. It is, of course, not possible to describe every conceivable combination of components or methodologies for purposes of describing the claimed subject matter, but one of ordinary skill in the art may recognize that many further combinations and permutations of the subject innovation are possible. Accordingly, the claimed subject matter is intended to embrace all such alterations, modifications, and variations that fall within the spirit and scope of the appended claims.

[0100] In particular and in regard to the various functions performed by the above described components, devices, circuits, systems and the like, the terms (including a reference to a “means”) used to describe such components are intended to correspond, unless otherwise indicated, to any component which performs the specified function of the described component (e.g., a functional equivalent), even though not structurally equivalent to the disclosed structure, which performs the function in the herein illustrated exemplary aspects of the claimed subject matter. In this regard, it will also be recognized that the innovation includes a system as well as a computer-readable storage media having computer-executable instructions for performing the acts and/or events of the various methods of the claimed subject matter.

[0101] There are multiple ways of implementing the subject innovation, e.g., an appropriate API, tool kit, driver code, operating system, control, standalone or downloadable software object, etc., which enables applications and services to use the techniques described herein. The claimed subject matter contemplates the use from the standpoint of an API (or other software object), as well as from a software or hardware

object that operates according to the techniques set forth herein. Thus, various implementations of the subject innovation described herein may have aspects that are wholly in hardware, partly in hardware and partly in software, as well as in software.

[00102] The aforementioned systems have been described with respect to interaction
5 between several components. It can be appreciated that such systems and components can include those components or specified sub-components, some of the specified components or sub-components, and/or additional components, and according to various permutations and combinations of the foregoing. Sub-components can also be implemented as components communicatively coupled to other components rather than included within
10 parent components (hierarchical).

[00103] Additionally, it can be noted that one or more components may be combined into a single component providing aggregate functionality or divided into several separate sub-components, and any one or more middle layers, such as a management layer, may be provided to communicatively couple to such sub-components in order to provide
15 integrated functionality. Any components described herein may also interact with one or more other components not specifically described herein but generally known by those of skill in the art.

[00104] In addition, while a particular feature of the subject innovation may have been disclosed with respect to only one of several implementations, such feature may be
20 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,” “including,” “has,” “contains,” variants thereof, and other similar words are used in either the detailed description or the claims, these terms are intended to be inclusive in a manner similar to the term “comprising” as an open transition word
25 without precluding any additional or other elements.

CLAIMS

What is claimed is:

1. A method for communicating with a client, comprising:
receiving a notification token that identifies the client;
5 registering the client to receive a message when the client is not connected to a server; and
sending the message to the client when the client does not have a connection to the server, based on the registering and the notification token, wherein the message is sent using a notification service.
- 10 2. The method recited in claim 1, wherein the notification channel comprises a hypertext transfer protocol secured (HTTPS) channel.
3. The method recited in claim 1, wherein registering the client comprises storing a notification channel uniform resource locator (URL) for the client.
4. The method recited in claim 3, comprising receiving a message specifying
15 the notification channel URL.
5. The method recited in claim 1, wherein the message is generated by a second client.
6. The method recited in claim 1, comprising subscribing the client to a buddy, wherein the buddy comprises a buddy client, wherein a notification channel is used
20 to send presence status updates of the buddy client to the client when the client is not connected.
7. The method recited in claim 6, comprising subscribing to one of:
a buddy list comprising one or more buddies;
a service; and
25 a network.
8. The method recited in claim 1, comprising sending a second message to the client corresponding to a shoulder tap, wherein the message comprises the shoulder tap, and wherein the second message comprises a payload.
9. A system for communicating with a client, comprising:
30 a processing unit; and
a system memory, wherein the system memory comprises code configured to direct the processing unit to:
receive a notification token that identifies the client;
register the client to receive a message when the client is not connected to a

server; and

send the message to the client when the client does not have a connection to the server, based on the registering and the notification token, wherein the message is sent using a notification channel.

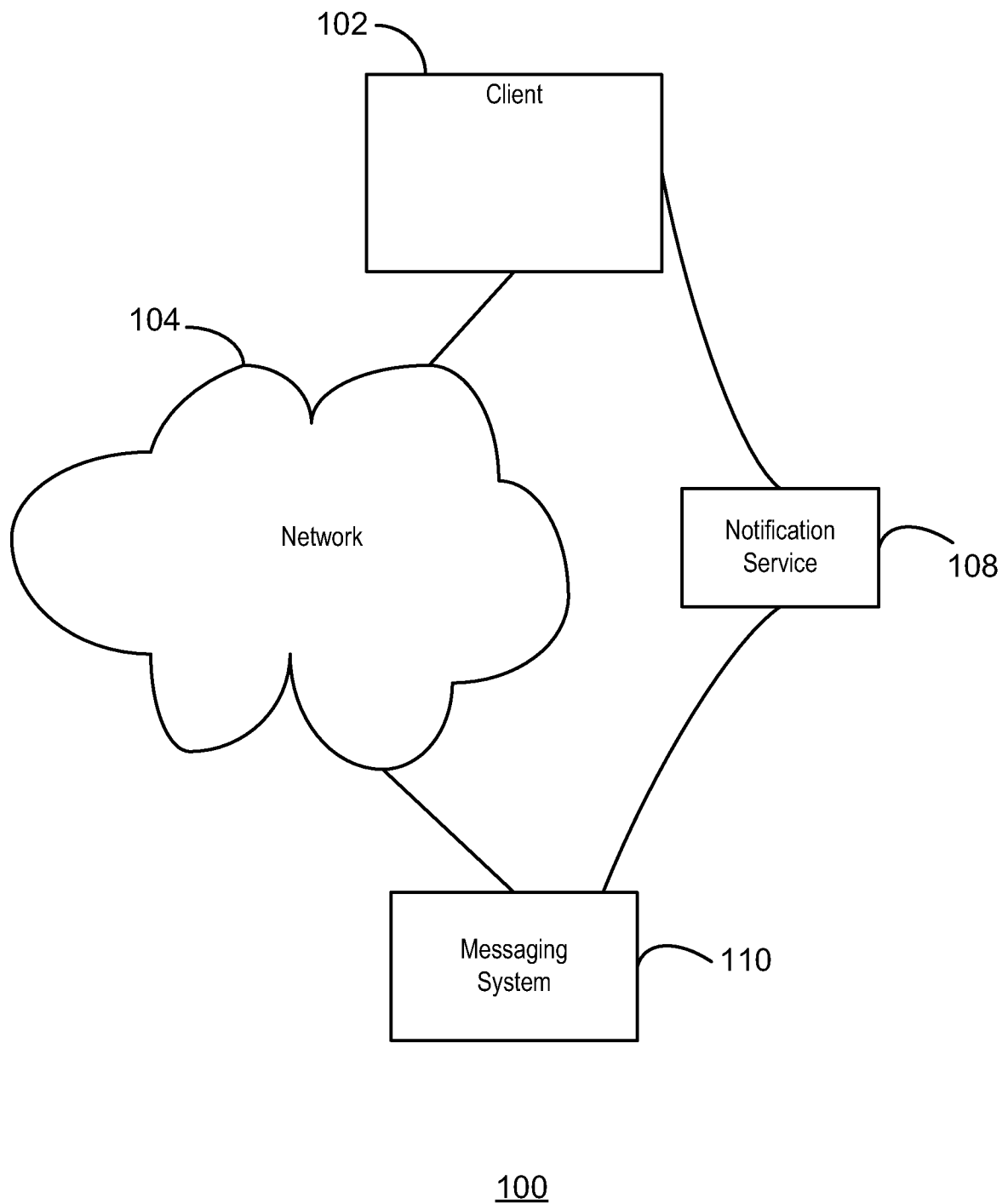
5 10. One or more computer-readable storage media, comprising code configured to direct a processing unit to:

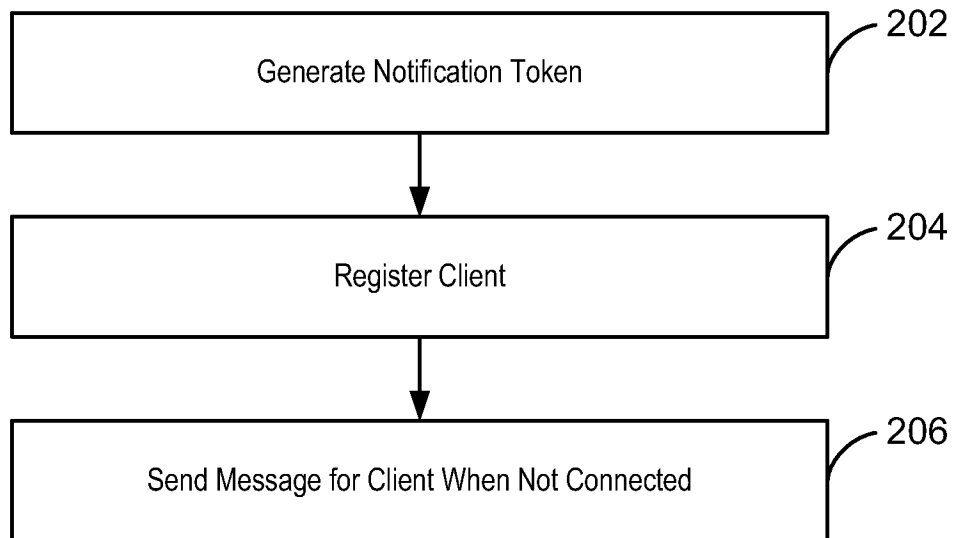
receive a notification token that identifies the client;

register the client to receive a message when the client is not connected to a server;

and

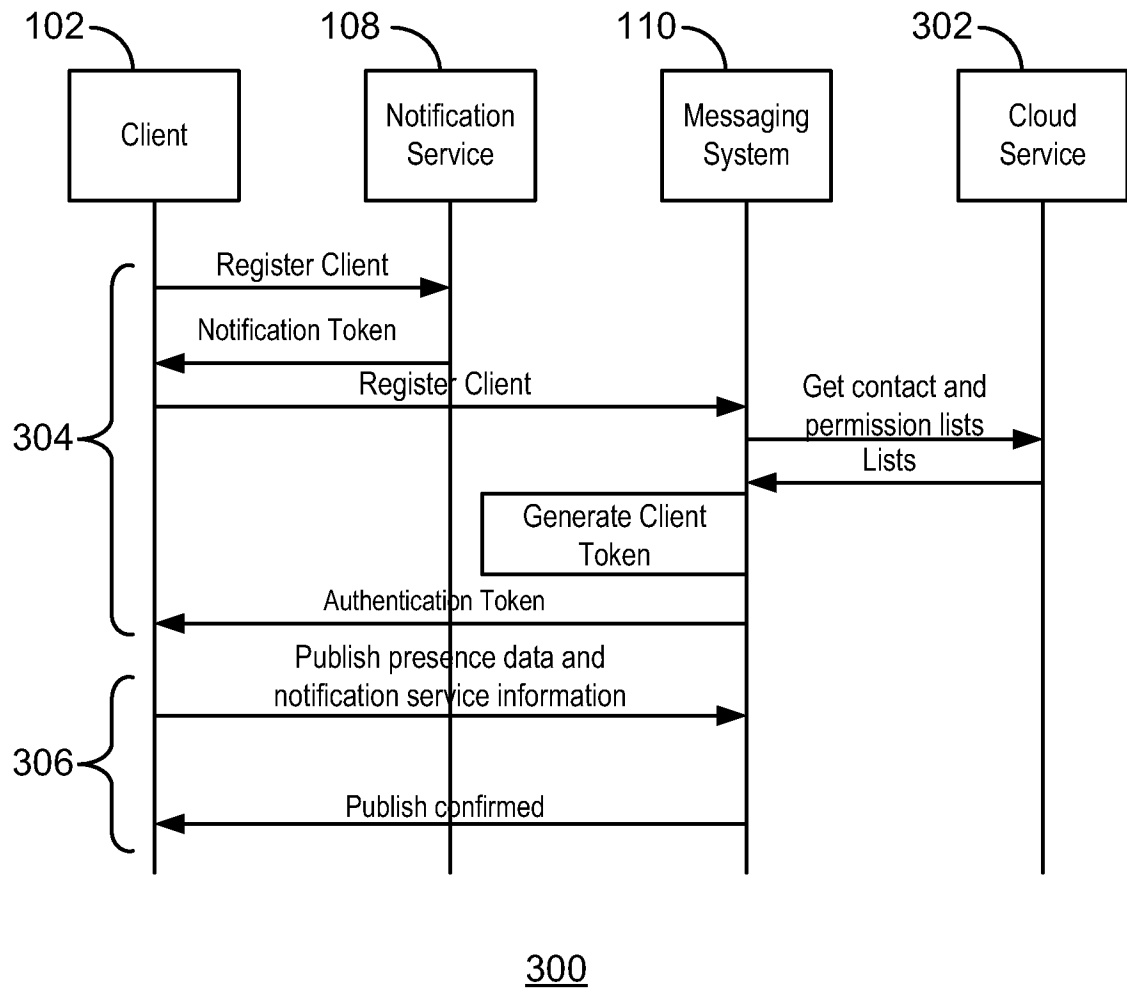
10 send the message to the client when the client does not have a connection to the server, based on the registering and the notification token, wherein the message is sent using a notification channel.

**FIG. 1**



200

FIG. 2

**FIG. 3**

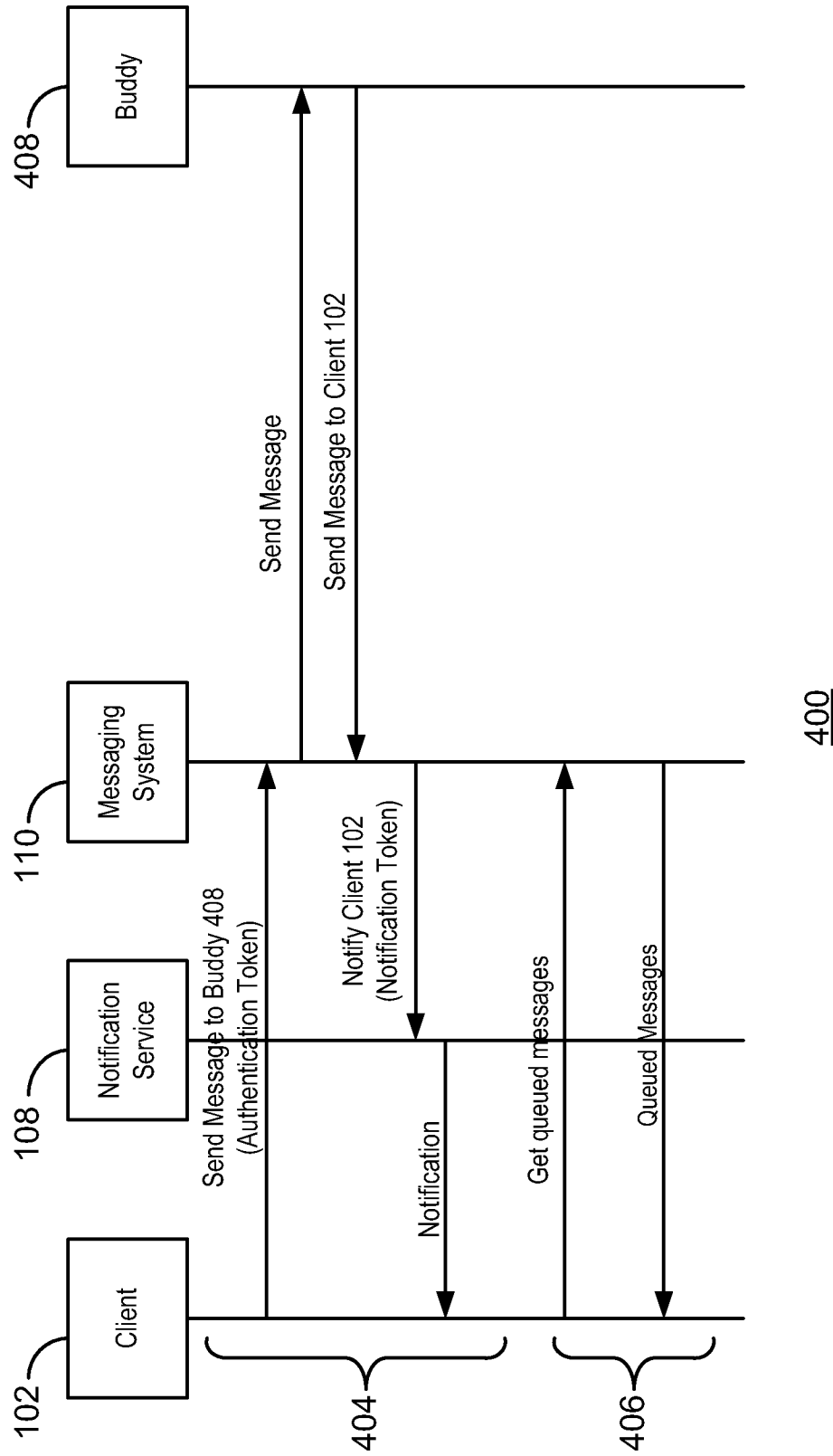


FIG. 4

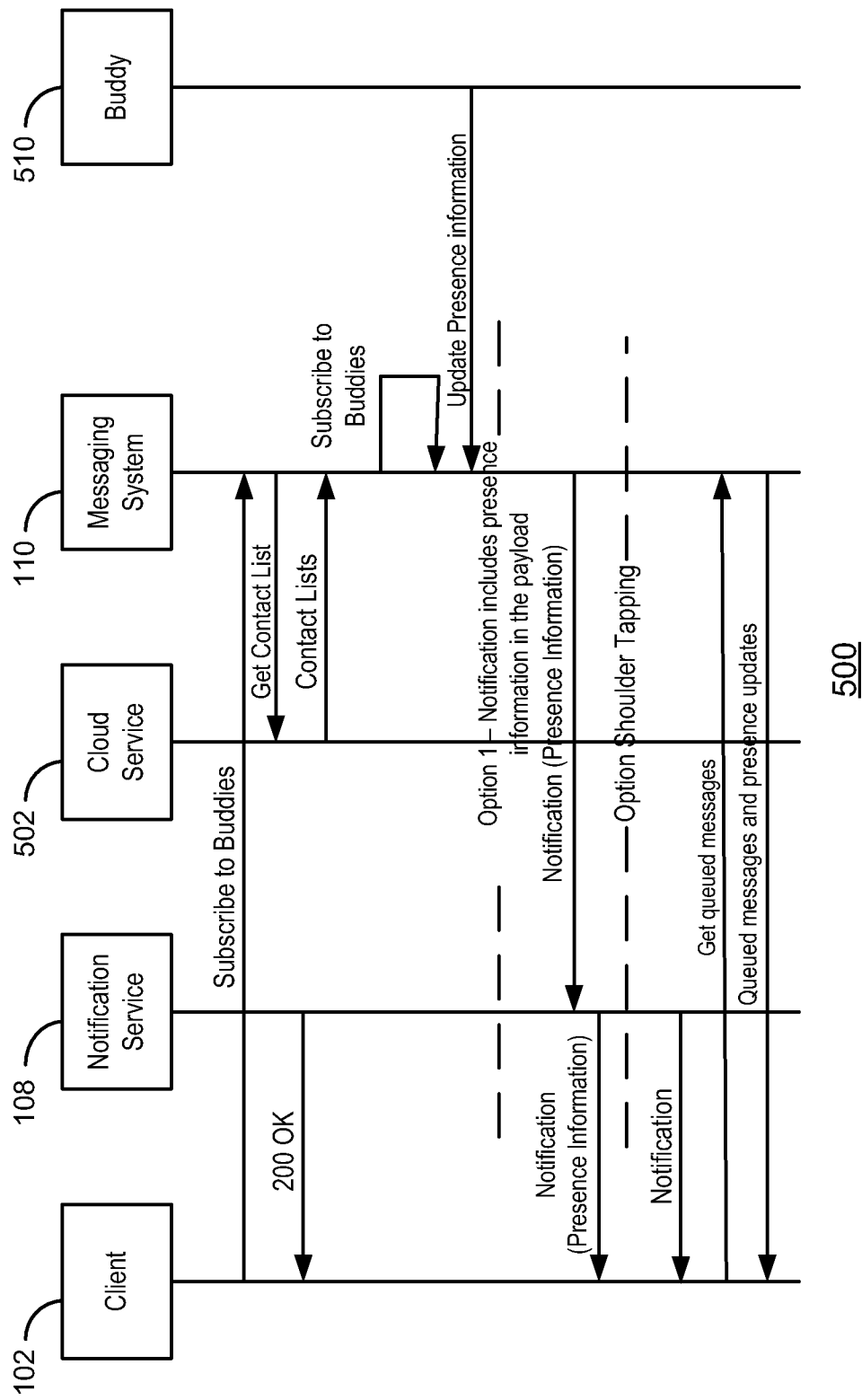
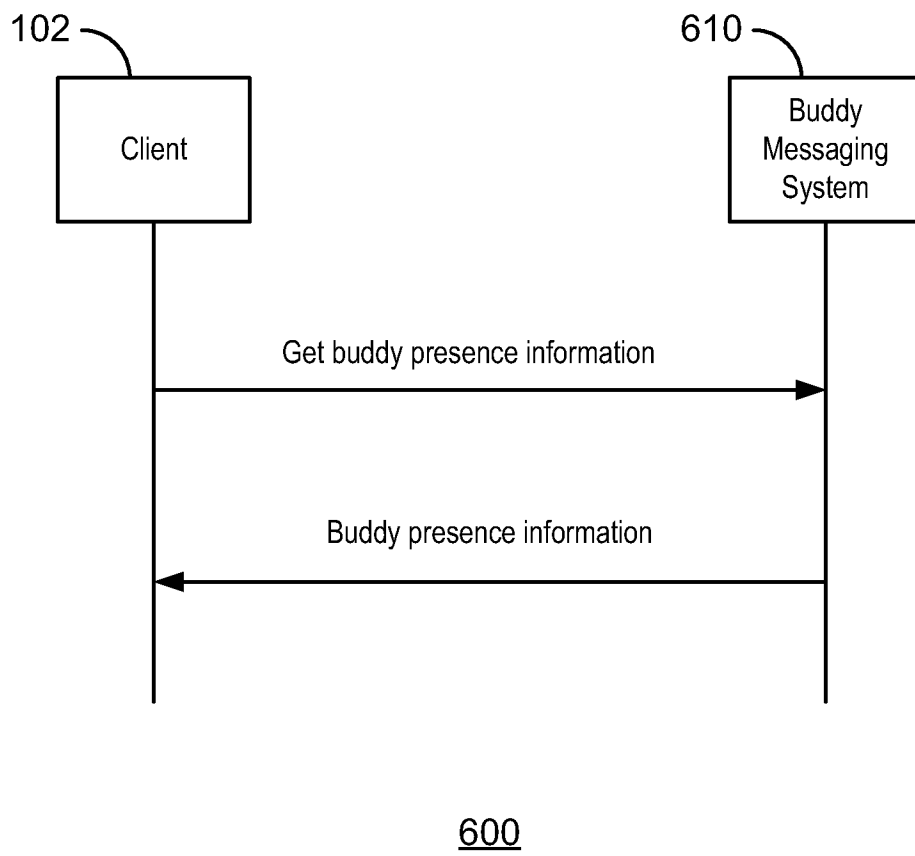
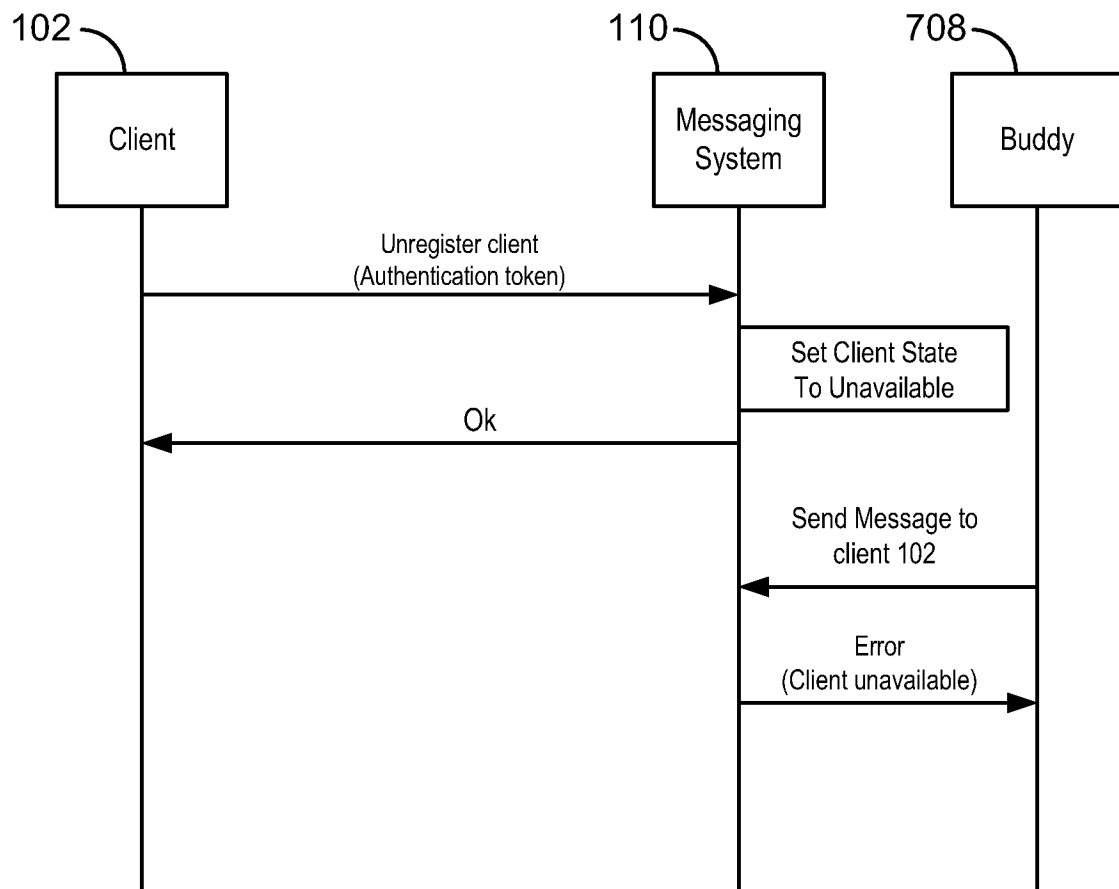
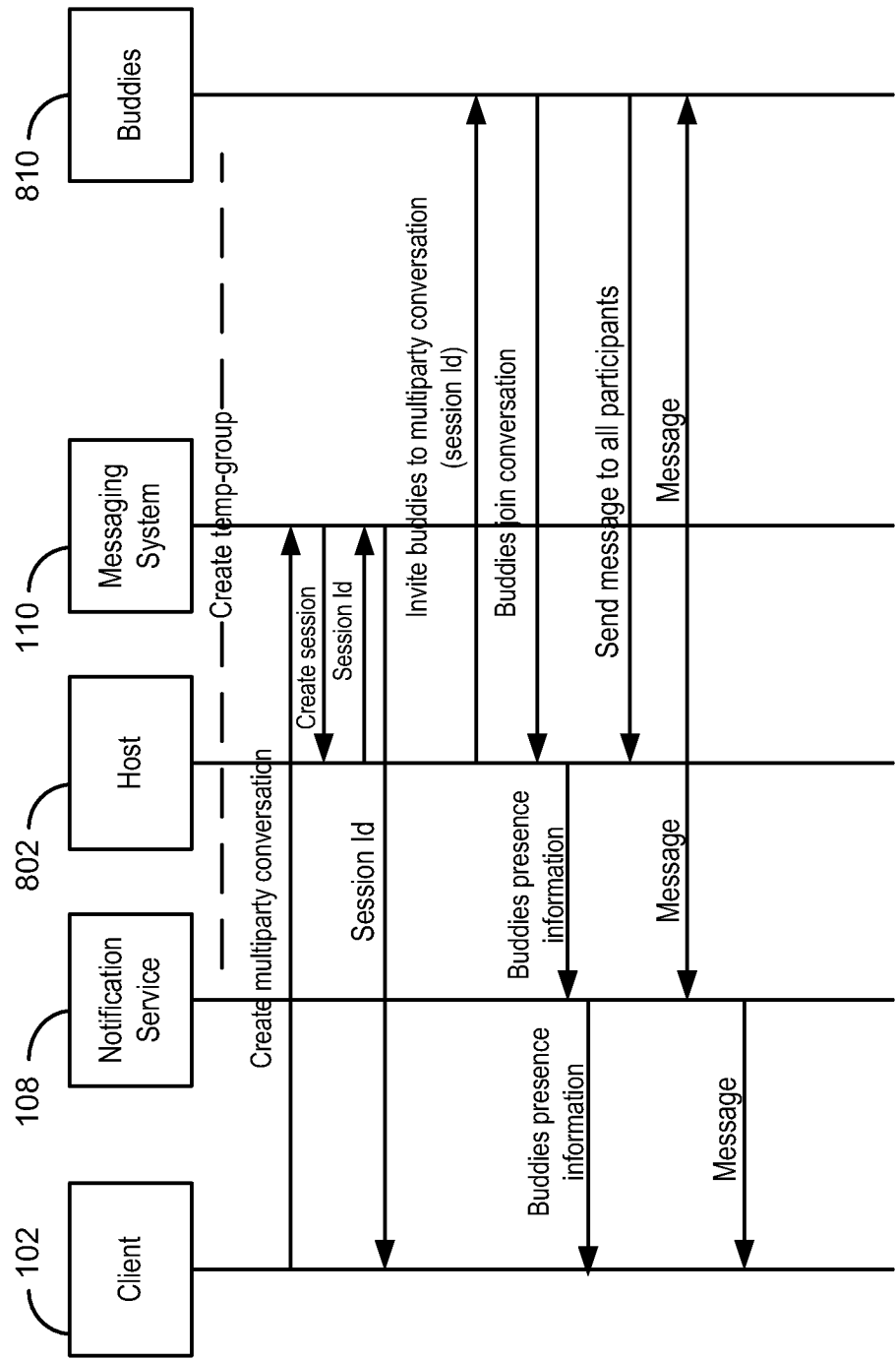


FIG. 5

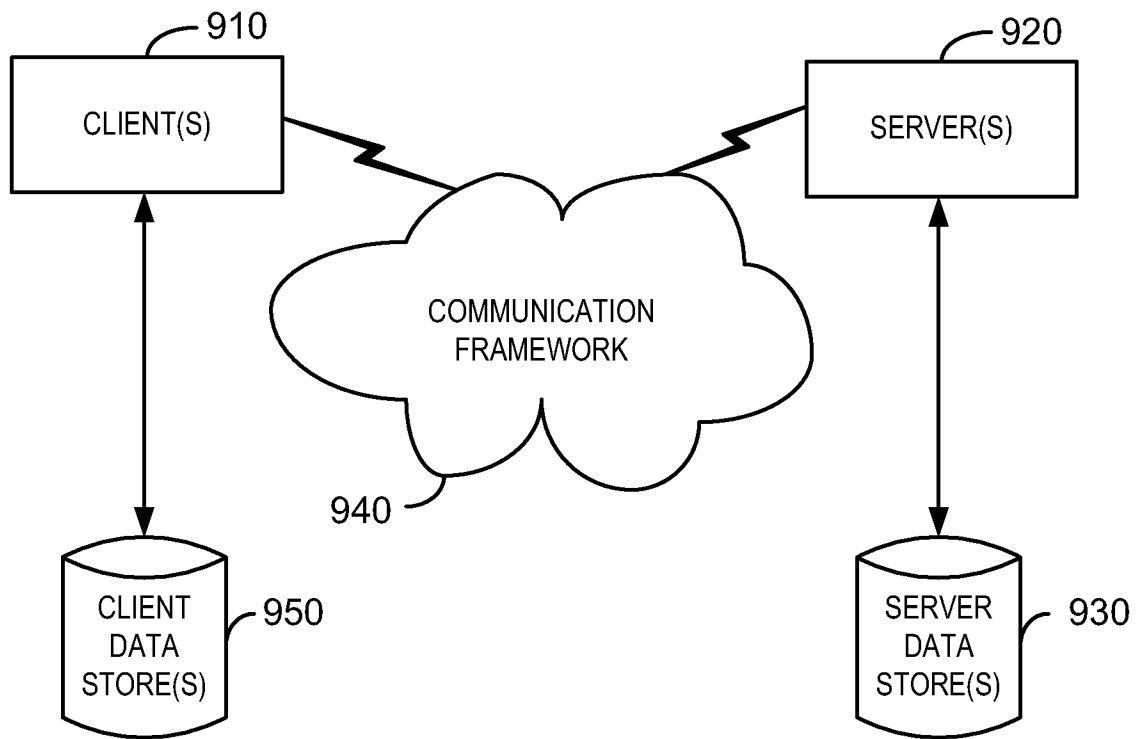
**FIG. 6**

700**FIG. 7**



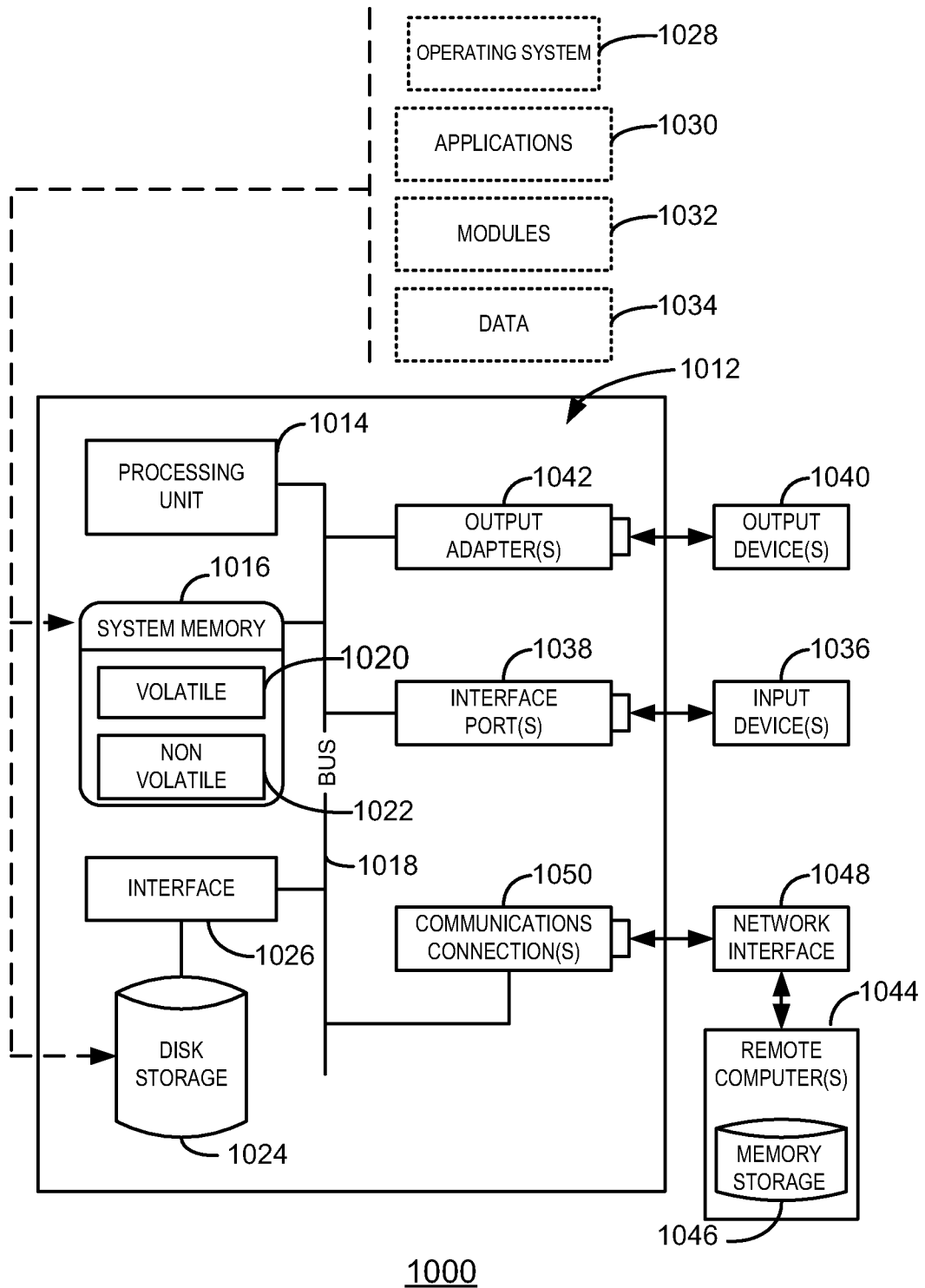
800

FIG. 8



900

FIG. 9

**FIG. 10**