



(51) International Patent Classification:

H04L 29/08 (2006.01) G06Q 10/10 (2012.01)

(21) International Application Number:

PCT/US2018/062376

(22) International Filing Date:

23 November 2018 (23.11.2018)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

15/835,279 07 December 2017 (07.12.2017) US

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

(72) Inventors: GUPTA, Sunitha; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Red-

mond, Washington 98052-6399 (US). SINGH, Jaskaran; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) Agent: MINHAS, Sandip S. et al.; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, 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.

(54) Title: AUTOMATIC SUBSCRIPTION MANAGEMENT OF COMPUTING SERVICES

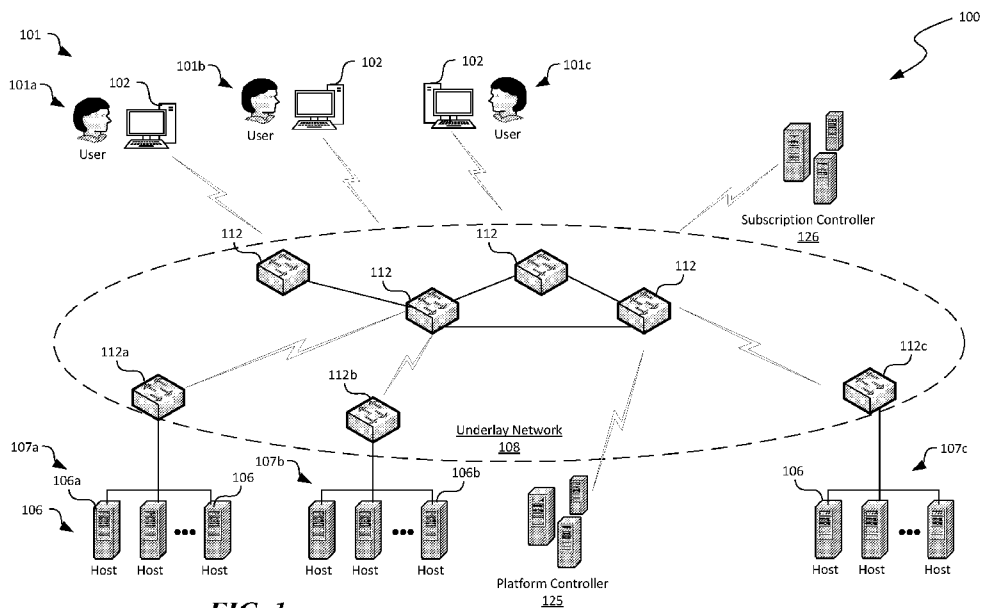


FIG. 1

(57) Abstract: Techniques of automatic subscription control in distributed computing systems are disclosed herein. In one embodiment, a method includes using a server to monitor a utilization level of an automatically subscribed computing service by a user. The method also includes determining whether the utilization level of the user with the automatically subscribed computing service is below a preset threshold. When the utilization level of the user is below the preset threshold, the server automatically unsubscribes the user from the computing service. While the computing service remains automatically unsubscribed to by the user, the server can monitor for a new version or update of the computing service. Upon detecting a new version or update, the server can automatically re-subscribe the user to the computing service of the new version without user input.



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

Declarations under Rule 4.17:

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

Published:

- *with international search report (Art. 21(3))*

AUTOMATIC SUBSCRIPTION MANAGEMENT OF COMPUTING SERVICES

BACKGROUND

5 [0001] Remote or “cloud” computing typically utilizes a collection of virtual machines or other remote servers, rather than local servers or personal computers, to process, manage, or store user data. A communications network can interconnect the remote servers as nodes to form a computing fabric. During operation, one or more nodes of the computing fabric can cooperate with one another to provide a distributed computing environment that facilitates execution of various software applications to provide corresponding computing
10 services.

SUMMARY

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

[0003] Remote computing can offer a wide variety of computation, storage, or other suitable types of computing services to users based on service subscription. For example, a user can subscribe to a hosted email service provided by a remote email server that is managed by a service provider. Based on the service subscription of the user, the user has access to emails,
20 calendar, reminders, tasks, and other items hosted on the remote email server. For example, the remote email server can facilitate the user to setup meetings by transmitting meeting invitations via emails.

[0004] From time to time, the service provider can deploy new services, update existing services, features, or otherwise alter the computing services available to the user. For
25 example, the service provider can deploy a new conversational scheduling service to users subscribed to the hosted email service. The conversation scheduling service can, with user permission, monitor user conversations in emails for detecting a meeting intent among users. In response to a detected meeting intent, the conversational scheduling service can be configured to automatically generate and transmit a meeting suggestion (e.g., as an email)
30 to the users. The meeting suggestion can include an email that provides access to one or more actions, for instance, generating an electronic meeting invitation for scheduling the meeting. As such, the conversational scheduling service can simplify meeting scheduling and improve efficiency of arranging for such meetings.

[0005] Though new features or computing services can potentially offer improved user

experience, the inventors have recognized that sometimes, early versions of new features or computing services can be rather limited in functionality, user friendliness, or have other shortcomings. As such, users experiencing the early versions may opt to unsubscribe from such new features or computing services due to negative user experience. Once
5 unsubscribed, the users may not experience any later versions of the features or computing services even though the later versions have additional functionalities, improved user friendliness, better designed user interfaces, or other functional improvements.

[0006] Several embodiments of the disclosed technology can address at least some aspects of the foregoing difficulties by implementing automatic subscription control of features or
10 computing services without user intervention. For example, in certain embodiments, once a new feature or computing service is initialized and provided to the users, a subscription controller can monitor for user interaction or utilization of the provided new feature or computing service. Thus, for the example conversational scheduling feature above, the subscription controller can be configured to detect whether a user receiving such meeting
15 suggestions have acted on the meeting suggestions. If the user has not acted on a certain number (e.g., three or four) of the meeting suggestions, the subscription controller can automatically unsubscribe the user from the conversational scheduling feature, without user input. In other embodiments, the foregoing subscription decision can also be based on an elapsed time since the user received a meeting suggestion, or based on other suitable
20 conditions.

[0007] The subscription controller can also be configured to monitor for new versions, updates, or other suitable modifications to features or computing services after the subscription controller has automatically unsubscribed the user. In response to detecting a new version of a feature or computing service, the subscription controller can determine
25 whether to restart the feature or computing service for the user. For example, if an update is simply a bug fix to the feature or computing service, the subscription controller can determine not to restart the feature of computing service. On the other hand, if the update is a new release of the feature or computing service, the subscription controller can decide to restart the feature or computing service. Thus, in response to determining that the feature or
30 computing service should be restarted, the subscription controller can automatically re-subscribe the user to the feature or computing service of the new version, without user input. The subscription controller can then continue to monitor for user interaction as discussed above.

[0008] Several embodiments of the disclosed technology can thus allow users to experience

new features or computing services without user intervention. Unlike in other computing systems, the subscription controller can automatically detect user interaction with a new feature or computing service. Based on the detected user interaction, the subscription controller can deduce whether a user is interested in using such a new feature or computing service. If the user is deemed not interested in the new feature or computing service, the subscription controller can automatically unsubscribe the user from the new feature or computing service. However, later on, if a new and improved version of the feature or computing service (e.g., with an improved user interface) becomes available, the subscription controller can automatically re-subscribe the user to the feature or computing service again, without user input. As such, several embodiments of the disclosed technology can allow the service provider to provide a better user experience to any features or computing services accessible to the users.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Figure 1 is a schematic diagram illustrating a distributed computing system implementing automatic subscription control in accordance with embodiments of the disclosed technology.

[0010] Figure 2 is a schematic diagram illustrating certain hardware/software components of the distributed computing system of Figure 1 in accordance with embodiments of the disclosed technology.

[0011] Figures 3A-3C are block diagrams of certain hardware/software components of the distributed computing system of Figure 1 during certain stages of automatic subscription control in the distributed computing system of Figure 1 in accordance with embodiments of the disclosed technology.

[0012] Figures 4A-4C are flowcharts illustrating processes of automatic subscription control in a distributed computing system in accordance with embodiments of the disclosed technology.

[0013] Figure 5 is a computing device suitable for certain components of the distributed computing system in Figure 1.

DETAILED DESCRIPTION

[0014] Certain embodiments of systems, devices, components, modules, routines, data structures, and processes for automatic subscription control for computing services provided by datacenters or other suitable distributed computing systems are described below. In the following description, specific details of components are included to provide a thorough understanding of certain embodiments of the disclosed technology. A person skilled in the

relevant art will also understand that the technology can have additional embodiments. The technology can also be practiced without several of the details of the embodiments described below with reference to Figures 1-5.

5 [0015] As used herein, the term “distributed computing system” generally refers to an interconnected computer network having a plurality of network nodes that connect a plurality of servers or hosts to one another or to external networks (e.g., the Internet). The term “network node” generally refers to a physical network device. Example network nodes include routers, switches, hubs, bridges, load balancers, security gateways, or firewalls. A “host” generally refers to a physical computing device configured to implement, for
10 instance, one or more virtual machines or other suitable virtualized components. For example, a host can include a server having a hypervisor configured to support one or more virtual machines or other suitable types of virtual components.

[0016] A computer network can be conceptually divided into an overlay network implemented over an underlay network. An “overlay network” generally refers to an
15 abstracted network implemented over and operating on top of an underlay network. The underlay network can include multiple physical network nodes interconnected with one another. An overlay network can include one or more virtual networks. A “virtual network” generally refers to an abstraction of a portion of the underlay network in the overlay network. A virtual network can include one or more virtual end points referred to as “tenant
20 sites” individually used by a user or “tenant” to access the virtual network and associated computing, storage, or other suitable resources. A tenant site can host one or more tenant end points (“TEPs”), for example, virtual machines. The virtual networks can interconnect multiple TEPs on different hosts. Virtual network nodes in the overlay network can be connected to one another by virtual links individually corresponding to one or more network
25 routes along one or more physical network nodes in the underlay network.

[0017] Also used herein, the term “computing service,” “service,” or “feature” generally refers to one or more computing resources provided over a computer network such as the Internet. Example cloud services include software as a service (“SaaS”), platform as a
30 service (“PaaS”), and infrastructure as a service (“IaaS”). SaaS is a software distribution technique in which software applications are hosted by a cloud service provider in, for instance, datacenters, and accessed by users over a computer network. PaaS generally refers to delivery of operating systems and associated services over the computer network without requiring downloads or installation. IaaS generally refers to outsourcing equipment used to support storage, hardware, servers, network devices, or other components, all of which are

made accessible over a computer network.

[0018] Also used herein, the term “subscription” or to “subscribe” to a computing service generally refers to obtaining a licensed access to a SaaS typically delivered via a computer network, such as the Internet. One example subscription can include Office365® provided
5 by Microsoft Corporation of Redmond, Washington. To “unsubscribe” from a computing service generally refers to declining the licensed access to the computing service. Further used herein, an “interaction level” with a computing service generally refers to a level of user actions of interfacing or utilizing the computing service. Example user actions can include providing user input to the computing service as well as consuming output (e.g.,
10 messages) from the computing service.

[0019] Service providers can sometimes deploy new services, update existing services, or otherwise alter the computing services available to users based on service subscription. For example, the service provider can deploy a new version of a computing service or add a feature to an existing computing service. Though new features or computing services can
15 potentially offer improved user experience, early versions of new features or computing services can sometimes be limited in functionality, user friendliness, or have other shortcomings. As such, users experiencing the early versions may opt to unsubscribe from such new features or computing services. Once unsubscribed, the users may not experience the same features or computing services even when later versions have additional
20 functionalities, improved user friendliness, or other improvements.

[0020] Several embodiments of the disclosed technology can improve user experience of computing services by implementing automatic subscription control of features or computing services. In certain embodiments, once a new feature or computing service is automatically subscribed to by a user, a subscription controller can monitor for user
25 interaction or utilization of the provided new feature or computing service. If the user has not acted on or utilize the computing service, the subscription controller can automatically unsubscribe the user from the computing service, without user input.

[0021] The subscription controller can also be configured to monitor for new versions, updates, or other suitable modifications to features or computing services that the
30 subscription controller has automatically unsubscribed for the user. In response to detecting a new version of a feature or computing service, the subscription controller can decide to re-subscribe the user to the computing service of the new version. Thus, several embodiments of the disclosed technology can allow users to experience new features or computing services without user intervention, as described in more detail below with

reference to Figures 1-5. In the description below, the automatic subscription control technique is described in the context of a distributed computing system. However, in other implementations, embodiments of the automatic subscription control can also be applied to computing services on a single computing device, or to computing services in other suitable computing environment.

[0022] Figure 1 is a schematic diagram illustrating a distributed computing system 100 implementing automatic subscription control in accordance with embodiments of the disclosed technology. As shown in Figure 1, the distributed computing system 100 can include an underlay network 108 interconnecting a plurality of hosts 106, a plurality of client devices 102 associated with corresponding users 101, a platform controller 125, and a subscription controller 126 operatively coupled to one another. Even though particular components of the distributed computing system 100 are shown in Figure 1, in other embodiments, the distributed computing system 100 can also include additional and/or different components or arrangements. For example, in certain embodiments, the subscription controller 126 can be an integral part of the platform controller 125, or one or more of the hosts 106 providing, for instance, hosted email services. In other embodiments, the distributed computing system 100 can also include network storage devices, additional hosts, and/or other suitable components (not shown).

[0023] As shown in Figure 1, the underlay network 108 can include one or more network nodes 112 that interconnect the multiple hosts 106, the users 101, and the platform controller 125. In certain embodiments, the hosts 106 can be organized into racks, action zones, groups, sets, or other suitable divisions. For example, in the illustrated embodiment, the hosts 106 are grouped into three host sets identified individually as first, second, and third host sets 107a-107c. In the illustrated embodiment, each of the host sets 107a-107c is operatively coupled to a corresponding network nodes 112a-112c, respectively, which are commonly referred to as “top-of-rack” or “TOR” network nodes. The TOR network nodes 112a-112c can then be operatively coupled to additional network nodes 112 to form a computer network in a hierarchical, flat, mesh, or other suitable types of topology. The computer network can allow communication between hosts 106, the platform controller 125, and the users 101. In other embodiments, the multiple host sets 107a-107c may share a single network node 112 or can have other suitable arrangements.

[0024] The hosts 106 can individually be configured to provide computing, storage, and/or other suitable cloud or other suitable types of computing services to the users 101 based on user subscriptions. For example, as described in more detail below with reference to Figure

2, one of the hosts 106 can initiate and maintain one or more virtual machines 144 (shown in Figure 2) upon requests from the users 101 when the users 101 have subscribed to such computing services. The users 101 can then utilize the initialized virtual machines 144 to perform computation, communication, and/or other suitable tasks, such as email hosting, meeting scheduling, etc. In certain embodiments, one of the hosts 106 can provide virtual machines 144 for multiple users 101. For example, the host 106a can host three virtual machines 144 individually corresponding to each of the users 101a-101c. In other embodiments, multiple hosts 106 can host virtual machines 144 for the users 101a-101c. As described in more detail herein, in certain implementations, certain computing services may be automatically subscribed to by the users 101, and the subscription controller 126 can be configured to manage such subscriptions automatically without user intervention.

[0025] The client devices 102 can each include a computing device that facilitates corresponding users 101 to access cloud services provided by the hosts 106 via the underlay network 108. For example, in the illustrated embodiment, the client devices 102 individually include a desktop computer. In other embodiments, the client devices 102 can also include laptop computers, tablet computers, smartphones, or other suitable computing devices. Even though three users 101 are shown in Figure 1 for illustration purposes, in other embodiments, the distributed computing system 100 can facilitate any suitable number of users 101 to access cloud or other suitable types of computing services provided by the hosts 106.

[0026] The platform controller 125 can be configured to manage operations of various components of the distributed computing system 100. For example, the platform controller 125 can be configured to allocate virtual machines 144 (or other suitable resources) in the distributed computing system 100, monitor operations of the allocated virtual machines 144, or terminate any allocated virtual machines 144 once operations are complete. In certain examples, the platform controller 125 can also be configured to monitor and track versions, updates, or other modifications to applications and corresponding computing services available in the distributed computing system 100. In the illustrated implementation, the platform controller 125 is shown as an independent hardware/software component of the distributed computing system 100. In other embodiments, the platform controller 125 can also be a datacenter controller, a fabric controller, or other suitable types of controller or a component thereof implemented as a computing service on one or more of the hosts 106.

[0027] The subscription controller 126 can be configured to automatically manage subscriptions to various features or computing services by the users 101 in the distributed

computing system 100. For example, in certain implementations, the subscription controller 126 can be configured to monitor a utilization level of a computing service by a user 101 after the computing device is automatically subscribed to by the user. The subscription controller 126 can also be configured to determine whether the utilization level of the user 5 101 with the automatically subscribed computing service is below a preset threshold. In response to determining that the utilization level of the user is below the preset threshold, the subscription controller 126 can automatically unsubscribe the user 101 from the computing service. While the computing service remains automatically unsubscribed to by the user 101, the subscription controller 126 can be configured to detect a new version or 10 update of the computing service is now available in the distributed computing system 100. Upon detecting the new version or update, the subscription controller 126 can automatically re-subscribe the user 101 to the computing service of the new version. As such, the user 101 can experience new features or computing services without user intervention, as described in more detail below with reference to Figures 3A-3C.

15 **[0028]** Figure 2 is a schematic diagram illustrating certain hardware/software components of the distributed computing system 100 in accordance with embodiments of the disclosed technology. In particular, Figure 2 illustrates an overlay network 108' that can be implemented on the underlay network 108 in Figure 1. Though particular configuration of the overlay network 108' is shown in Figure 2, in other embodiments, the overlay network 20 108' can also be configured in other suitable ways. In Figure 2, only certain components of the underlay network 108 of Figure 1 are shown for clarity.

[0029] In Figure 2 and in other Figures herein, individual software components, objects, classes, modules, and routines may be a computer program, procedure, or process written as source code in C, C++, C#, Java, and/or other suitable programming languages. A 25 component may include, without limitation, one or more modules, objects, classes, routines, properties, processes, threads, executables, libraries, or other components. Components may be in source or binary form. Components may include aspects of source code before compilation (e.g., classes, properties, procedures, routines), compiled binary units (e.g., libraries, executables), or artifacts instantiated and used at runtime (e.g., objects, processes, 30 threads).

[0030] Components within a system may take different forms within the system. As one example, a system comprising a first component, a second component and a third component can, without limitation, encompass a system that has the first component being a property in source code, the second component being a binary compiled library, and the

third component being a thread created at runtime. The computer program, procedure, or process may be compiled into object, intermediate, or machine code and presented for execution by one or more processors of a personal computer, a network server, a laptop computer, a smartphone, and/or other suitable computing devices.

5 **[0031]** Equally, components may include hardware circuitry. A person of ordinary skill in the art would recognize that hardware may be considered fossilized software, and software may be considered liquefied hardware. As just one example, software instructions in a component may be burned to a Programmable Logic Array circuit, or may be designed as a hardware circuit with appropriate integrated circuits. Equally, hardware may be emulated
10 by software. Various implementations of source, intermediate, and/or object code and associated data may be stored in a computer memory that includes read-only memory, random-access memory, magnetic disk storage media, optical storage media, flash memory devices, and/or other suitable computer readable storage media excluding propagated signals.

15 **[0032]** As shown in Figure 2, the first host 106a and the second host 106b can each include a processor 132, a memory 134, and an input/output component 136 operatively coupled to one another. The processor 132 can include a microprocessor, a field-programmable gate array, and/or other suitable logic devices. The memory 134 can include volatile and/or nonvolatile media (e.g., ROM; RAM, magnetic disk storage media; optical storage media;
20 flash memory devices, and/or other suitable storage media) and/or other types of computer-readable storage media configured to store data received from, as well as instructions for, the processor 132 (e.g., instructions for performing the methods discussed below with reference to Figures 4A and 4B). The input/output component 136 can include a display, a touch screen, a keyboard, a mouse, a printer, and/or other suitable types of input/output
25 devices configured to accept input from and provide output to an operator and/or an automated software controller (not shown).

[0033] The first and second hosts 106a and 106b can individually contain instructions in the memory 134 executable by the processors 132 to cause the individual processors 132 to provide a hypervisor 140 (identified individually as first and second hypervisors 140a and
30 140b) and a status agent 141 (identified individually as first and second status agent 141a and 141b). Even though the hypervisor 140 and the status agent 141 are shown as separate components, in other embodiments, the status agent 141 can be a part of the hypervisor 140 or an operating system (not shown) executing on the corresponding host 106. In further embodiments, the status agent 141 can be a standalone application.

[0034] The hypervisors 140 can individually be configured to generate, monitor, terminate, and/or otherwise manage one or more virtual machines 144 organized into tenant sites 142. For example, as shown in Figure 2, the first host 106a can provide a first hypervisor 140a that manages first and second tenant sites 142a and 142b, respectively. The second host
5 106b can provide a second hypervisor 140b that manages first and second tenant sites 142a' and 142b', respectively. The hypervisors 140 are individually shown in Figure 2 as a software component. However, in other embodiments, the hypervisors 140 can be firmware and/or hardware components. The tenant sites 142 can each include multiple virtual machines 144 for a particular tenant (not shown). For example, the first host 106a and the
10 second host 106b can both host the tenant site 142a and 142a' for a first tenant 101a (Figure 1). The first host 106a and the second host 106b can both host the tenant site 142b and 142b' for a second tenant 101b (Figure 1). Each virtual machine 144 can be executing a corresponding operating system, middleware, and/or applications 147.

[0035] Also shown in Figure 2, the distributed computing system 100 can include an overlay
15 network 108' having one or more virtual networks 146 that interconnect the tenant sites 142a and 142b across multiple hosts 106. For example, a first virtual network 142a interconnects the first tenant sites 142a and 142a' at the first host 106a and the second host 106b. A second virtual network 146b interconnects the second tenant sites 142b and 142b' at the first host 106a and the second host 106b. Even though a single virtual network 146 is
20 shown as corresponding to one tenant site 142, in other embodiments, multiple virtual networks 146 (not shown) may be configured to correspond to a single tenant site 146.

[0036] The virtual machines 144 can be configured to execute one or more applications 147 to provide suitable cloud or other suitable types of computing services to the users 101 (Figure 1). The virtual machines 144 on the virtual networks 146 can also communicate
25 with one another via the underlay network 108 (Figure 1) even though the virtual machines 144 are located on different hosts 106. Communications of each of the virtual networks 146 can be isolated from other virtual networks 146. In certain embodiments, communications can be allowed to cross from one virtual network 146 to another through a security gateway or otherwise in a controlled fashion. A virtual network address can correspond to one of the
30 virtual machine 144 in a particular virtual network 146. Thus, different virtual networks 146 can use one or more virtual network addresses that are the same. Example virtual network addresses can include IP addresses, MAC addresses, and/or other suitable addresses.

[0037] Figures 3A-3C are block diagrams of certain hardware/software components of the distributed computing system 100 of Figure 1 during certain stages of automatic

subscription control in the distributed computing system 100 of Figure 1 in accordance with embodiments of the disclosed technology. In Figures 3A-3C, only one host 106 and the subscription controller 126 are shown for illustration purposes.

5 [0038] As shown in Figure 3A, the host 106 can execute suitable instructions to provide a virtual machine 144 in which an application 147 is executed to provide a corresponding computing service. In the illustrated embodiment, the computing service can include one that generates one or more service messages 162 transmitted to the client device 102 for a user 101 via the underlay/overlay network 108 and 108'. Example service messages 162 can include emails, electronic notifications, simple text messages, or other suitable types of
10 messages.

[0039] In one example, the computing service provided to the user 101 can include a conversational scheduling service. The conversation scheduling service can, with permission from the user 101, monitor user conversations in emails for detecting a meeting intent among users 101. In response to a detected meeting intent, the conversational
15 scheduling service can be configured to automatically generate and transmit a service message 162 in the form of a meeting suggestion to the user 101. The meeting suggestion can include an email that provides access to one or more actions, for instance, generating an electronic meeting invitation for scheduling the meeting. As such, the conversational scheduling service can simplify meeting scheduling and improve efficiency of arranging for
20 such meetings.

[0040] In another example, the computing service provided to the user 101 can include a notification service for upcoming agenda items collected, with permission of the user 101, from an electronic calendar associated with the user 101. In such an example, the service message 162 can include an email containing data representing upcoming appointments,
25 scheduled meetings, or other suitable calendar items. In further examples, the computing service provided to the user 101 can also include services for map direction suggestion, current driving time notification, or other suitable computing services. Though only one application 147 is shown in Figures 3A-3C for illustration purposes, in other embodiments, the host 106 can also execute multiple applications 147 (not shown) in one or more virtual
30 machines 144.

[0041] As shown in Figure 3A, the host 106 can also execute additional instructions to provide a reporting agent 145 configured to generate interaction data 158 related to the provided computing service. The reporting agent 145 can be a part of the operating system, hypervisor 140 (Figure 2), or other suitable components on the host 106. In one

embodiment, the reporting agent 145 can monitor any interactions by the user 101 with the service messages 162. For example, the reporting agent 145 can track whether the user 101 has clicked on a link included in the service messages 162 to, for instance, generate and transmit a suggested meeting invitation. In another example, the reporting agent 145 can also track whether the user 101 has clicked on any of the calendar items to retrieve further information. In other examples, the reporting agent 145 can detect whether the user 101 has read the service messages 162, deleted the service messages 162 with or without reading, or other suitable actions taken by the user 101.

[0042] Once the interaction data 158 is generated, the reporting agent 145 can also provide the generated interaction data 158 to the subscription controller 126 periodically or in other suitable manners. Such reporting can be upon request from the subscription controller 126, without any prompt from the subscription controller 126, or in other suitable manners. Even though the reporting agent 145 is described above as being configured to generate and report the interaction data 158, in other embodiments, the host 106 may provide the generating and reporting functionalities related to the interaction data 158 using separate applications, modules, or routines.

[0043] As shown in Figure 3A, the subscription controller 126 can be operatively coupled to a computer readable storage 151. The subscription controller 126 can also include a data collector 152, a data analyzer 154, and an auto subscriber 156 operatively coupled to one another. Even though particular components and configurations of the subscription controller 126 are shown in Figures 3A-3C, in other embodiments, the subscription controller 126 can also include other suitable components or arrangements. For example, in certain embodiments, the computer readable storage 151 can be an internal non-volatile storage of a computing device implementing the subscription controller 126. In other examples, the computer readable storage 151 can also be a network storage space in the distributed computing system 100.

[0044] The computer readable storage 151 can contain records of interaction data 158 and application data 159. The interaction data 158 can include data representing a level of utilization or interaction with a computing service provided by one or more hosts 106 (Figure 1) in the distributed computing system 100. In the illustrated embodiment in Figure 3A, interaction data 158 is provided to the subscription controller 126 from the reporting agent 145 of host 106. In other embodiments, the platform controller 125 (Figure 1) or other suitable components of the distributed computing system 100 can be configured to provide the interaction data 158 or other suitable types of data to the subscription controller 126.

[0045] The application data 159 can include data representing a version, update, build, or other suitable edition information related to an application 147 configured to provide a corresponding computing service. In one example, the application data 159 can contain a series of integers delineated by periods, such as, “10.1.16,” in which “10” designates a main
5 version number while “1” and “16” designate a subversion number and a build number, respectively. In other examples, the application data 159 can also contain other suitable types of numbers or alphanumerical values that represent edition information of the application 147. In further examples, the application data 159 can also contain comments such as “bug fix,” “user interface update,” or other suitable details of modifications in
10 addition to or in lieu of the edition information.

[0046] The data collector 152 can be configured to collect interaction data 158 from one or more applications 147 or the virtual machines 144 hosted on the host 106. In certain embodiments, the data collector 152 can query the host 106 for the interaction data 158 regarding utilization of the computing service corresponding to the application 147
15 periodically or in other suitable manners. In turn, the data collector 152 can be configured to receive the requested interaction data 158 from the reporting agent 145. In other embodiments, the data collector 152 can simply receive the interaction data 158 from the reporting agent 145 without any prompt. In further embodiments, the data collector 152 can be configured to implement a combination of the foregoing data retrieving techniques. The
20 data collector 152 can then provide the received interaction data 158 to the data analyzer 154 for further processing and store the interaction data 158 in the computer readable storage 151.

[0047] The data analyzer 154 can be configured to process the received interaction data 158 based on a preset interaction threshold. In certain embodiments, the data analyzer 154 can
25 be configured to aggregate the received interaction data 158 related to the application 147 and obtain a total value of interaction level for a preset period (e.g., one week). In other embodiments, the data analyzer 154 can include a counter configured to obtain a number of service messages 162 transmitted to the user 101 and without interaction. In further embodiments, the data analyzer 154 can include routines for calculating an average, a
30 medium, an elapsed time, or other suitable parameters for measuring an interaction level of the user 101 with the provided computing service.

[0048] The data analyzer 154 can also be configured to compare the processed interaction data 158 with a preset interaction threshold and indicate whether sufficient interaction has been detected based thereon. In one example, the data analyzer 154 can be configured to

indicate that sufficient interaction has not been detected based on the following conditions:

- when a number of service messages 162 have been transmitted to the user 101 but no interaction by the user 101 has been detected;
- when a number of service messages 162 have been transmitted to the user 101 and have been deleted by the user 101 without being read; or
- when an elapsed time since one of the service messages 162 has been transmitted to the user 101 and remained without user interaction is above a present time threshold; or

In other examples, the data analyzer 154 can be configured to indicate whether sufficient interaction has been detected based other suitable criteria. Based on the foregoing determination, the data analyzer 154 can then indicate to the auto subscriber 156 regarding whether indicate whether sufficient interaction with the computing service has been detected.

[0049] The auto subscriber 156 can be configured to automatically subscribe and/or unsubscribe the user 101 to the provided computing service corresponding to the application 147 based on the indication from the data analyzer 154. As shown in Figure 3A, the auto subscriber 156 can automatically subscribe the user 101 to the provided computing service when the computing service is a new feature included in a service package according to a user subscription of the user 101, and that the computing service has not been automatically unsubscribed by the auto subscriber 156.

[0050] In another embodiment, the auto subscriber 156 can automatically unsubscribe the user 101 from the computing service when the data analyzer 154 indicates that sufficient interaction between the user 101 and the computing service has not been detected, as shown in Figure 3B. In the illustrated example, the auto subscriber 156 can be configured to transmit an unsubscribe instruction 164 to the application 147, the virtual machine 144, or the host 106 for unsubscribing the user 101 from the computing service. In another example, the auto subscriber 156 can be configured to set an automatic subscription flag associated with the computing service and store the set automatic subscription flag in an email inbox of the user 101 on the host 106. In response, the host 106 can stop executing the application 147 (shown in phantom lines) and/or perform other suitable operations such that no additional service messages 162 (shown in phantom lines) are transmitted to the user 101 when the automatic subscription flag is set or when the unsubscribe instruction 164 is received. In further examples, the auto subscriber 156 can be configured to unsubscribe the

user 101 from the computing service in other suitable manners.

[0051] Subsequently, a new version, update, or other suitable edition of the application 147' may be available on the host 106, as shown in Figure 3C. In the illustrated embodiment, the reporting agent 145 can transmit additional application data 159' to the data collector 152 of the subscription controller 126 indicating the latest version, update, or edition of the application 147'. In response, the data analyzer 154 can be configured to determine whether the new version or update of the application 147' represents a significant change in the computing service based on a preset rule. For example, the preset rule can indicate a significant change if a version number and/or a subversion number of the application 147' is different than the original application 147 (Figure 3A). In another example, the preset rule can indicate an insignificant change if the application data 159' includes a description that the update is only for bug fixes. In a further example, the preset rule can indicate a significant change if the application data 159' includes a description that the update contains a new user interface for the computing service. In yet further examples, the preset rule can indicate a significant change based on other suitable criteria.

[0052] As shown in Figure 3C, when the data analyzer 154 indicates to the auto subscriber 156 that the new version, update, or other edition information of the application 147' represents a significant change, the auto subscriber 156 can automatically re-subscribe the user 101 to the new version, update, or edition of the computing service without user intervention. For example, in the illustrated example, the auto subscriber 156 can be configured to transmit a subscribe instruction 166 to the host 106. In other examples, the auto subscriber 156 can be configured to reset the automatic subscription flag of the computing service and storing the reset automatic subscription flag in the email inbox of the user 101 on the host 106. In response, the host 106 can determine whether the automatic subscription flag of the computing service is set for the user 101. In response to determining that the automatic subscription flag of the computing service is not set for the user 101, re-initialize the computing service for the user 101.

[0053] Several embodiments of the distributed computing system 100 can thus allow the user 101 to experience new features or computing services without user intervention. Unlike in certain other computing systems, the subscription controller 126 can automatically subscribe the user 101 to and detect user interaction with a new feature or computing service. Based on the detected user interaction, the subscription controller 126 can deduce whether a user is interested in using such a new feature or computing service. If the user 101 is deemed not interested in the new feature or computing service due to various reasons, the

subscription controller 126 can automatically unsubscribe the user 101 from the new feature or computing service. However, later, if a new and improved version of the feature or computing service (e.g., with an improved user interface) becomes available, the subscription controller 126 can automatically re-subscribe the user 101 to the feature or computing service again, without user input. As such, the distributed computing system 100 can provide a better user experience to any features or computing services accessible to the user 101.

[0054] Figures 4A and 4B are flowcharts illustrating processes of automatic subscription control in a distributed computing system 100 in accordance with embodiments of the disclosed technology. Even though the processes are described in relation to the distributed computing system 100 of Figure 1 and 2 and the hardware/software components of Figures 3A-3C, in other embodiments, the processes can also be implemented in other suitable computing systems.

[0055] As shown in Figure 4A, a process 200 includes initializing a computing service for a user at stage 202. In one embodiment, the computing service can be automatically subscribed to by the user without user input when, for example, the computing service is a new feature included in a service package according to a service subscription of the user. Example operations of initializing the computing service are described in more detail below with reference to Figure 4B. The process 200 can then include monitoring user interaction with the provided computing service at stage 204. Various techniques may be used to determine an interaction level of the user to the computing service. Examples of such techniques are described above with reference to Figures 3A-3C. The process 200 can then include a decision stage 206 to determine whether sufficient interaction with the computing service by the user is detected. In response to determining that sufficient interaction with the computing service by the user is detected, the process 200 proceeds to continuing the computing service 208 before reverting to monitoring further user interaction with the computing service at stage 204. Otherwise, the process 200 proceeds to automatically unsubscribing the user from the computing service at stage 210. Example techniques for automatically unsubscribing the user from the computing service are described above with reference to Figures 3A-3C.

[0056] Figure 4B illustrates example operations for initializing a computing service for a user. As shown in Figure 4B, the operations can include detecting a service trigger at stage 212. A service trigger can include an indication of a new computing service or feature, or other suitable indicators. The operations can then include a decision stage 214 to determine

whether the user has been automatically unsubscribed from the computing service, for example, based on a current value of an automatic subscription flag stored in a mailbox of the user, as described in more detail above with reference to Figures 3A-3C. In response to determining that the user has been automatically unsubscribed from the computing service, the operations include ignoring the service trigger and terminating the initialization process at stage 218. Otherwise, the operations include executing a corresponding application to provide the computing service at stage 216.

[0057] Figure 4C illustrates a process 220 for automatically re-subscribing a user to a previously unsubscribed computing service. As shown in Figure 4C, the process 220 can include monitoring for a new edition notification regarding the computing service at stage 222. The process 220 can then include a decision stage 224 to determine whether to restart the subscription to the computing service. In certain embodiments, the process 220 can indicate to restart the computing service by determining whether a significant change in the computing service is detected based on the new edition notification, as described in more detail above with reference to Figure 3C. In other embodiments, the process 220 can restart the subscription when any new edition notification is received. In response to determining to restart the subscription, the process 220 can include automatically subscribe the user to the new edition of the computing device at stage 228. Example operations for automatically subscribe the user to the computing service are described in more detail above with reference to Figure 3C. Once restarted, the process 220 can further include monitoring for user interaction with the computing service, such as that described above with reference to Figure 4A. Otherwise, the process 220 can include updating application data 159 (Figure 3C) corresponding to the computing service before reverting to monitoring for additional new edition notifications at stage 222.

[0058] Figure 5 is a computing device 300 suitable for certain components of the distributed computing system 100 in Figure 1. For example, the computing device 300 can be suitable for the hosts 106, the client devices 102, the platform controller 125, or the subscription controller 126 of Figure 1. In a very basic configuration 302, the computing device 300 can include one or more processors 304 and a system memory 306. A memory bus 308 can be used for communicating between processor 304 and system memory 306.

[0059] Depending on the desired configuration, the processor 304 can be of any type including but not limited to a microprocessor (μ P), a microcontroller (μ C), a digital signal processor (DSP), or any combination thereof. The processor 304 can include one more levels of caching, such as a level-one cache 310 and a level-two cache 312, a processor core

314, and registers 316. An example processor core 314 can include an arithmetic logic unit (ALU), a floating point unit (FPU), a digital signal processing core (DSP Core), or any combination thereof. An example memory controller 318 can also be used with processor 304, or in some implementations memory controller 318 can be an internal part of processor 304.

[0060] Depending on the desired configuration, the system memory 306 can be of any type including but not limited to volatile memory (such as RAM), non-volatile memory (such as ROM, flash memory, etc.) or any combination thereof. The system memory 306 can include an operating system 320, one or more applications 322, and program data 324. As shown in Figure 5, the operating system 320 can include a hypervisor 140 for managing one or more virtual machines 144. This described basic configuration 302 is illustrated in Figure 8 by those components within the inner dashed line.

[0061] The computing device 300 can have additional features or functionality, and additional interfaces to facilitate communications between basic configuration 302 and any other devices and interfaces. For example, a bus/interface controller 330 can be used to facilitate communications between the basic configuration 302 and one or more data storage devices 332 via a storage interface bus 334. The data storage devices 332 can be removable storage devices 336, non-removable storage devices 338, or a combination thereof. Examples of removable storage and non-removable storage devices include magnetic disk devices such as flexible disk drives and hard-disk drives (HDD), optical disk drives such as compact disk (CD) drives or digital versatile disk (DVD) drives, solid state drives (SSD), and tape drives to name a few. Example computer storage media can include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules, or other data. The term “computer readable storage media” or “computer readable storage device” excludes propagated signals and communication media.

[0062] The system memory 306, removable storage devices 336, and non-removable storage devices 338 are examples of computer readable storage media. Computer readable storage media include, but not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other media which can be used to store the desired information and which can be accessed by computing device 300. Any such computer readable storage media can be a part of computing device 300. The term “computer readable storage medium” excludes propagated

signals and communication media.

5 [0063] The computing device 300 can also include an interface bus 340 for facilitating communication from various interface devices (e.g., output devices 342, peripheral interfaces 344, and communication devices 346) to the basic configuration 302 via bus/interface controller 330. Example output devices 342 include a graphics processing unit 348 and an audio processing unit 350, which can be configured to communicate to various external devices such as a display or speakers via one or more A/V ports 352. Example peripheral interfaces 344 include a serial interface controller 354 or a parallel interface controller 356, which can be configured to communicate with external devices such as input
10 devices (e.g., keyboard, mouse, pen, voice input device, touch input device, etc.) or other peripheral devices (e.g., printer, scanner, etc.) via one or more I/O ports 358. An example communication device 346 includes a network controller 360, which can be arranged to facilitate communications with one or more other computing devices 362 over a network communication link via one or more communication ports 364.

15 [0064] The network communication link can be one example of a communication media. Communication media can typically be embodied by computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transport mechanism, and can include any information delivery media. A “modulated data signal” can be a signal that has one or more of its characteristics set or
20 changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media can include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, radio frequency (RF), microwave, infrared (IR) and other wireless media. The term computer readable media as used herein can include both storage media and communication media.

25 [0065] The computing device 300 can be implemented as a portion of a small-form factor portable (or mobile) electronic device such as a cell phone, a personal data assistant (PDA), a personal media player device, a wireless web-watch device, a personal headset device, an application specific device, or a hybrid device that include any of the above functions. The computing device 300 can also be implemented as a personal computer including both
30 laptop computer and non-laptop computer configurations.

[0066] Specific embodiments of the technology have been described above for purposes of illustration. However, various modifications can be made without deviating from the foregoing disclosure. In addition, many of the elements of one embodiment can be combined with other embodiments in addition to or in lieu of the elements of the other embodiments.

Accordingly, the technology is not limited except as by the appended claims.

CLAIMS

- [c1]** 1. A method performed by a server in a distributed computing system having a plurality of servers interconnected by a computer network, the method comprising: initializing a computing service for a user, the initialized computing service being provided by one or more of the servers in the distributed computing system; upon initializing the computing service, receiving data representing an interaction level of the user with the automatically initialized computing service; analyzing the received data to determine whether the interaction level of the user is below a preset threshold; in response to determining that the interaction level of the user is below the preset threshold, automatically unsubscribing the user from the computing service; and while the computing service remains automatically unsubscribed to by the user, receiving a notification of a new version or update of the automatically unsubscribed computing service being available in the distributed computing system; and in response to receiving the notification, automatically re-initializing the computing service of the new version for the user in the distributed computing system without user input.
- [c2]** 2. The method of claim 1 wherein initializing the computing service for the user includes: detecting a meeting intent based on content in emails of the user; and upon detecting a meeting intent, generating and transmitting, via the computer network, an email with a meeting suggestion for scheduling a meeting with the user.
- [c3]** 3. The method of claim 1 wherein initializing the computing service for the user includes: determining whether the computing service has been previously automatically unsubscribed; and in response to determining that the computing service has not been previously automatically unsubscribed, instructing the one or more servers in the distributed computing system to execute an application configured to provide the computing service.

- [c4]** 4. The method of claim 1 wherein:
automatically unsubscribing the user includes setting an automatic subscription flag of the computing service and storing the set automatic subscription flag in an email inbox of the user on the one or more servers; and
initializing the computing service for the user includes:
determining whether the automatic subscription flag of the computing service is set for the user; and
in response to determining that the automatic subscription flag of the computing service is not set for the user, instructing the one or more servers in the distributed computing system to execute an application configured to provide the computing service.
- [c5]** 5. The method of claim 1 wherein initializing the computing service for the user includes:
automatically unsubscribing the user includes setting an automatic subscription flag of the computing service and storing the set automatic subscription flag in an email inbox of the user on the one or more servers; and
initializing the computing service for the user includes:
determining whether the automatic subscription flag of the computing service is set for the user; and
in response to determining that the automatic subscription flag of the computing service is set for the user, terminating initialization of the computing service in the distributed computing system.
- [c6]** 6. The method of claim 1 wherein:
initializing the computing service for the user includes providing the user with a plurality of service messages via email; and
receiving the data representing the interaction level of the user includes one of the following:
receiving data representing a number of the service messages transmitted to the user and remained without user interaction; or
receiving data representing an elapsed time since one of the service messages has been transmitted to the user and remained without user interaction.
- [c7]** 7. The method of claim 6 wherein analyzing the received data to determine whether the interaction level of the user includes at least one of:

determining whether the number of the service messages transmitted to the user and remained without user interaction is above a preset number threshold; or
determining whether the elapsed time since one of the service messages has been transmitted to the user and remained without user interaction is above a present time threshold.

[c8] 8. The method of claim 6 wherein:
analyzing the received data to determine whether the interaction level of the user includes at least one of:
determining whether the number of the service messages transmitted to the user and remained without user interaction is above a preset number threshold; or
determining whether the elapsed time since one of the service messages has been transmitted to the user and remained without user interaction is above a present time threshold; and
automatically unsubscribing the user includes setting an automatic subscription flag of the computing service and storing the set automatic subscription flag in an email inbox of the user on the one or more servers.

[c9] 9. The method of claim 1, further comprising:
in response to receiving the notification,
determining whether the new version or update represents a significant change in the computing service based on a preset rule; and
in response to determining that the new version or update represents a significant change in the computing service based on the preset rule, automatically re-initializing the computing service of the new version for the user in the distributed computing system without user input.

[c10] 10. A server in a distributed computing system having a plurality of additional servers interconnected by a computer network, the server comprising:
a processor; and
a memory containing instructions executable by the processor to cause the server to perform a process according to one of claims 1-9.

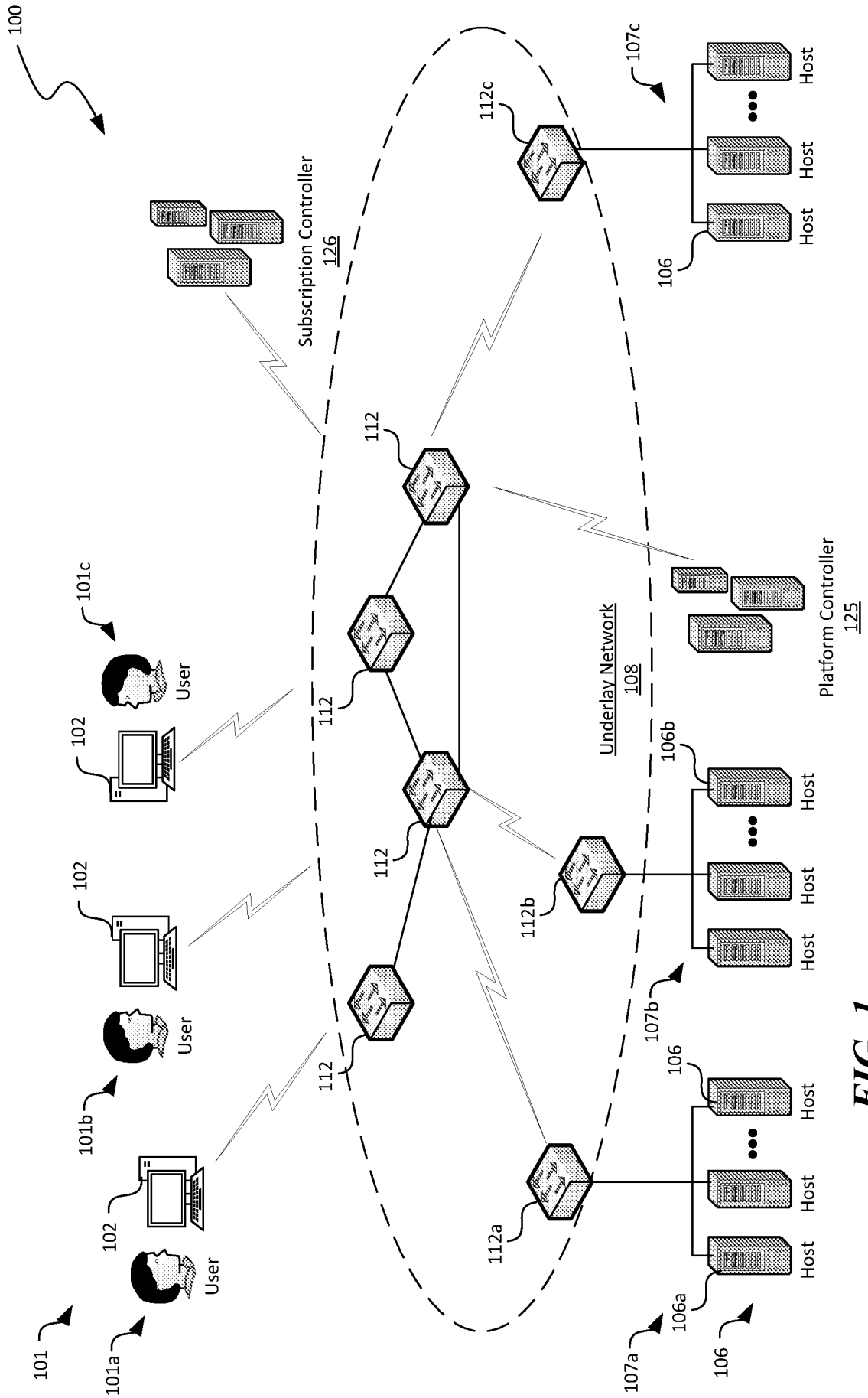


FIG. 1

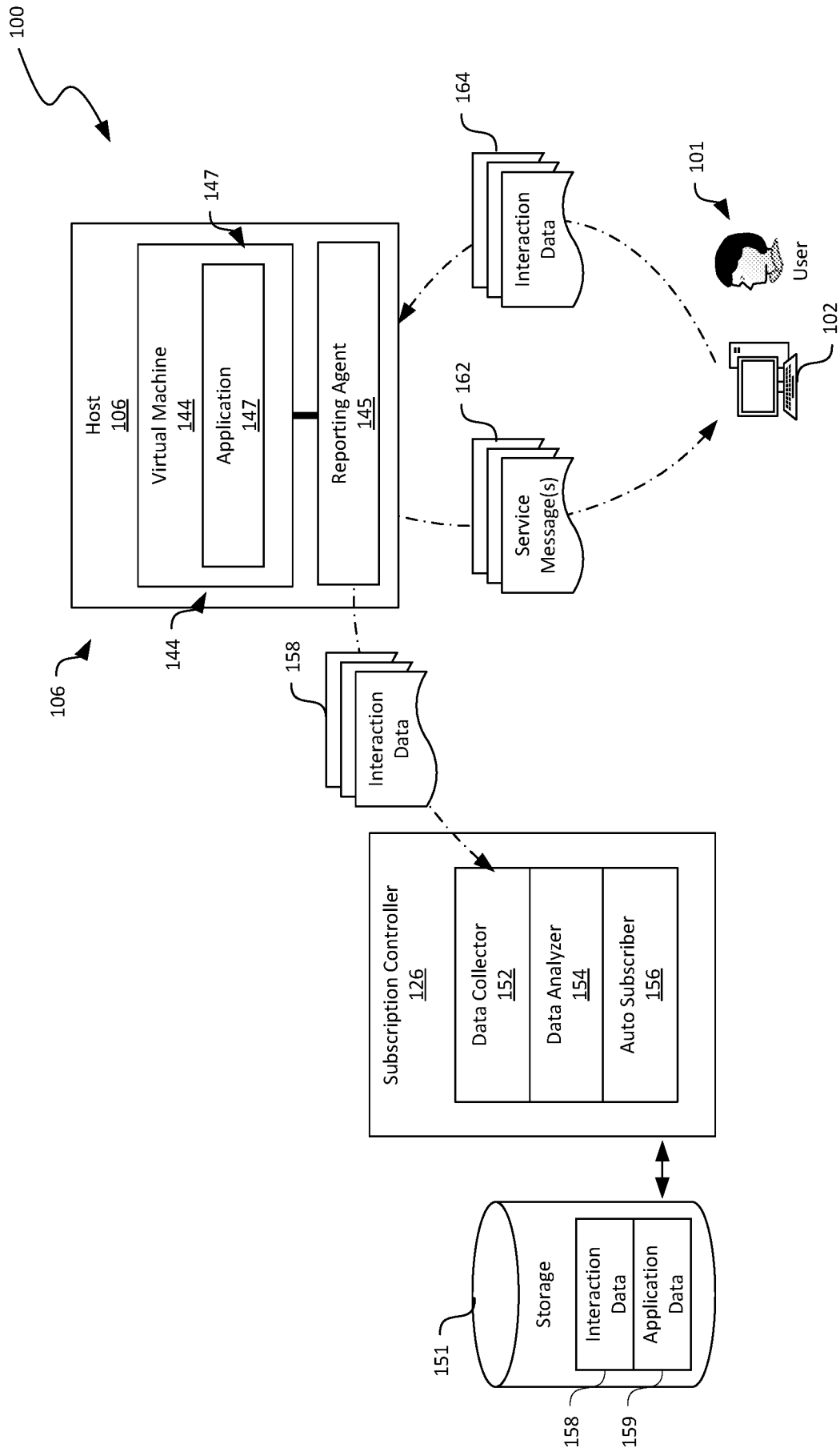


FIG. 3A

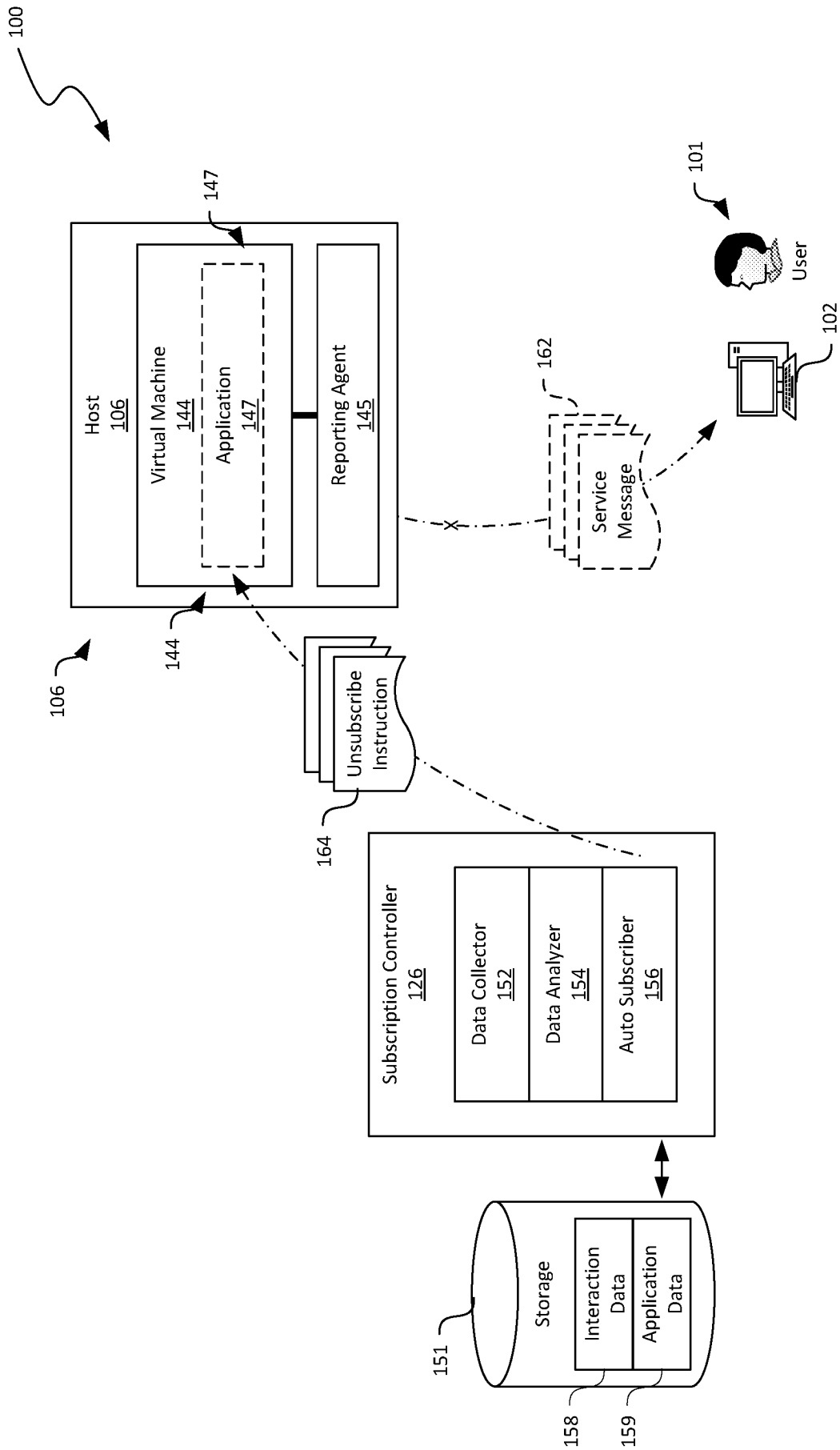


FIG. 3B

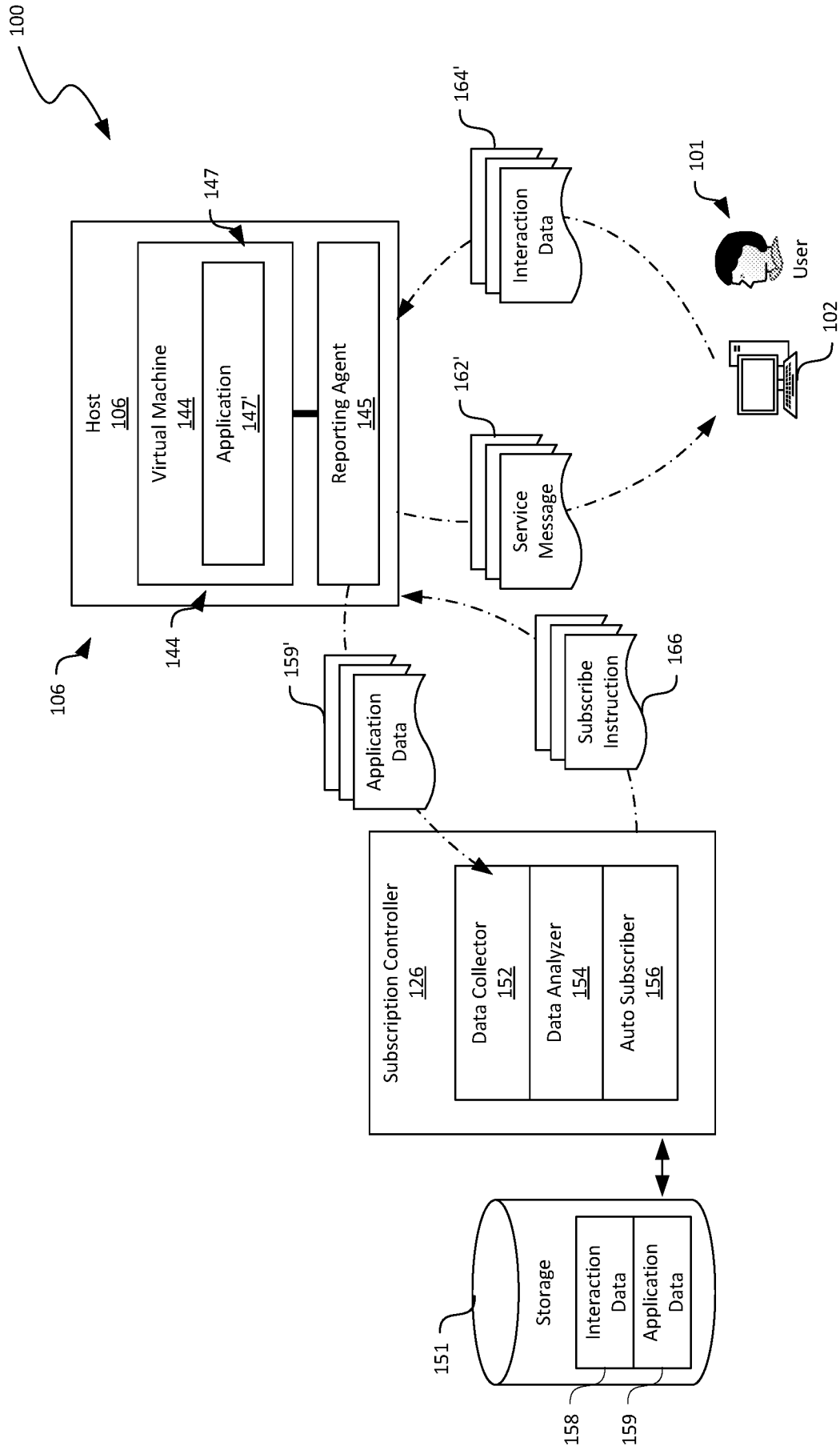


FIG. 3C

6/8

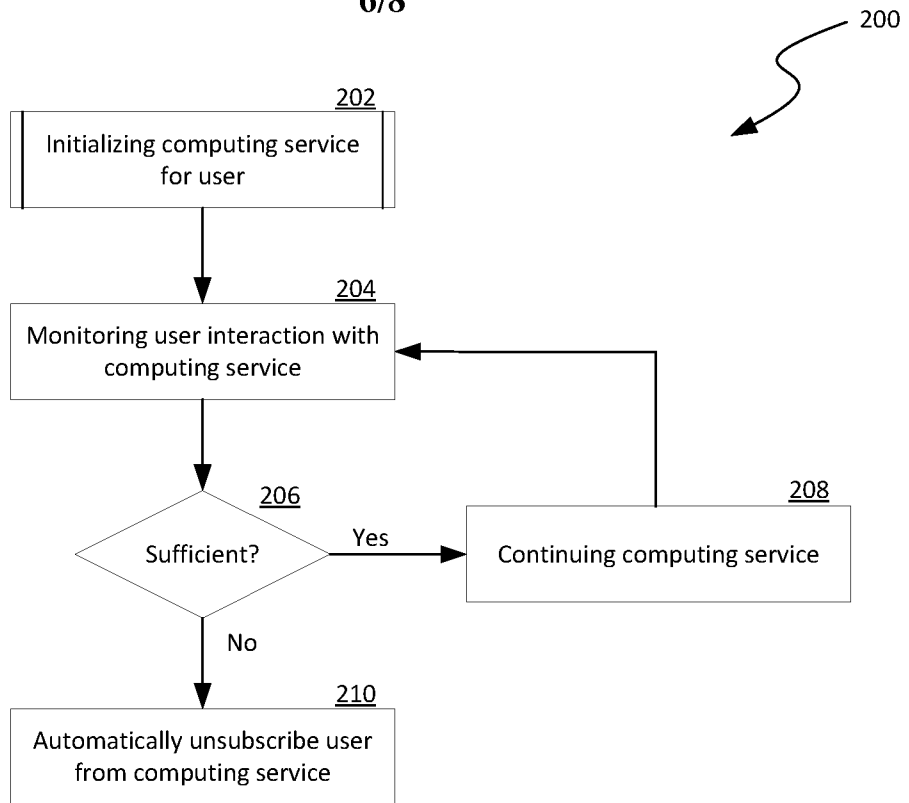


FIG. 4A

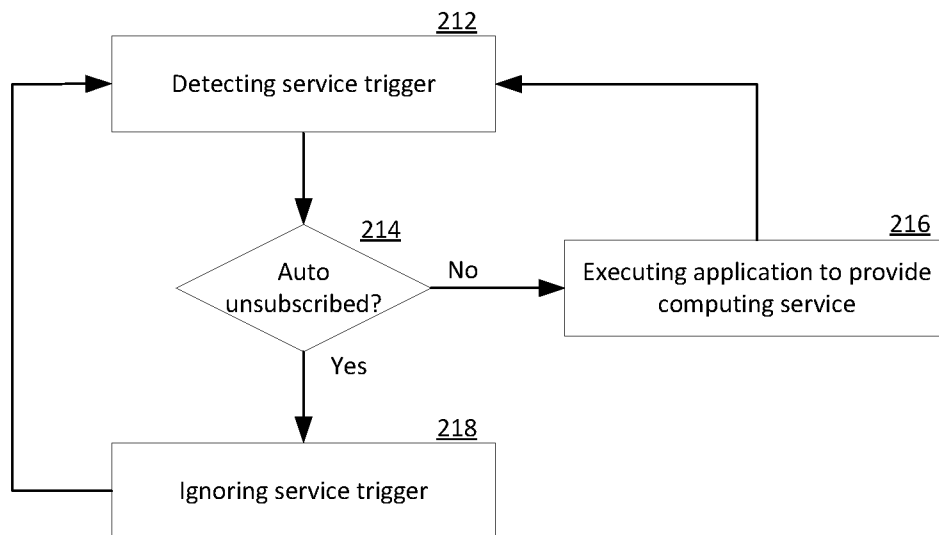


FIG. 4B

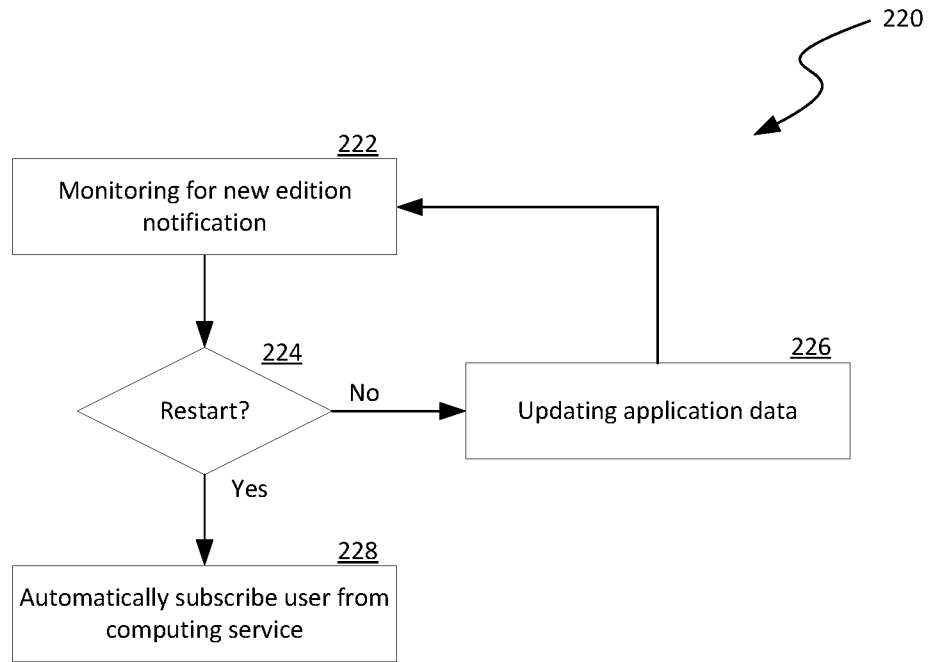
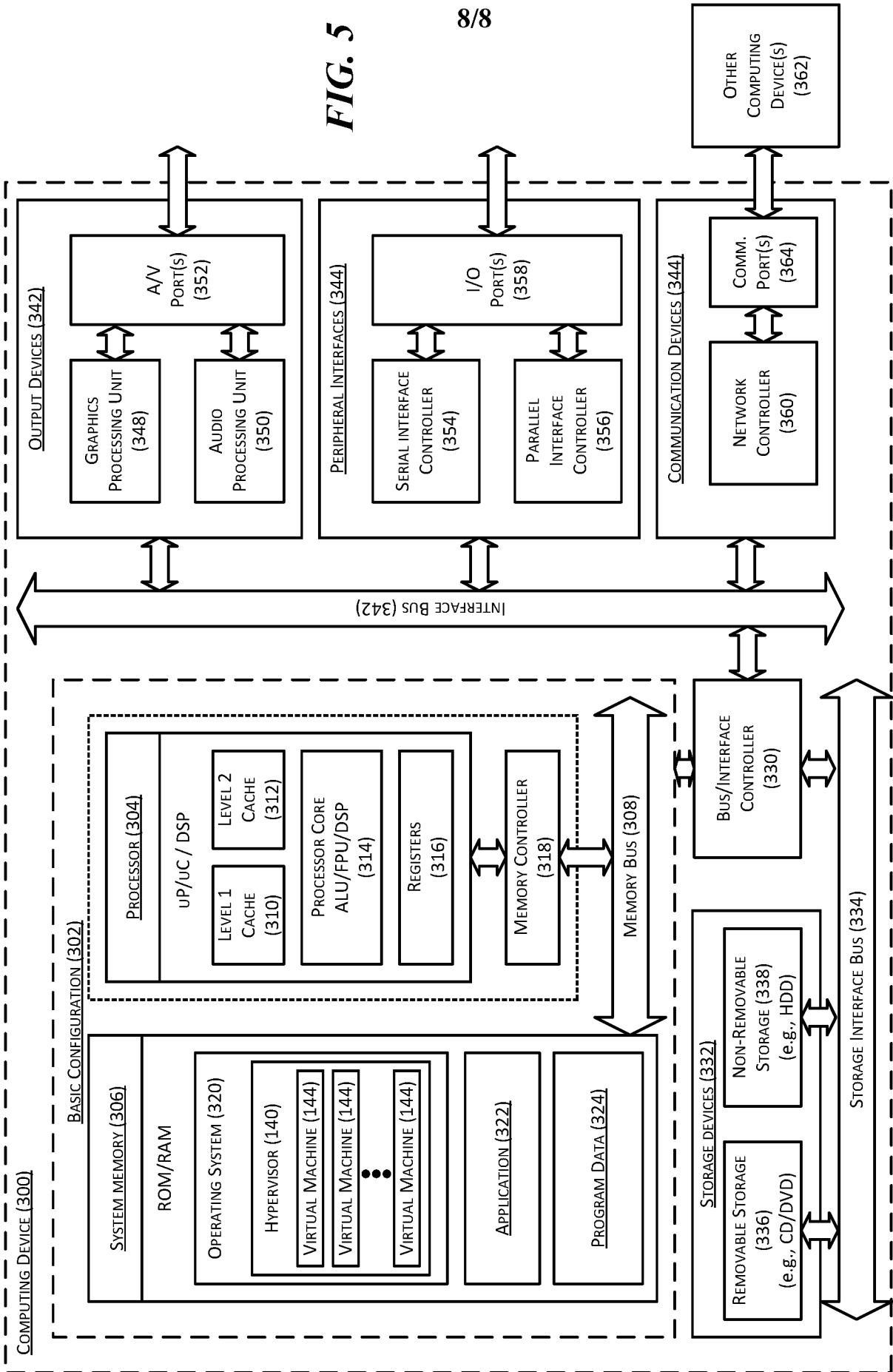


FIG. 4C

FIG. 5



INTERNATIONAL SEARCH REPORT

International application No
PCT/US2018/062376

A. CLASSIFICATION OF SUBJECT MATTER
 INV. H04L29/08 G06Q10/10
 ADD.
 According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
 Minimum documentation searched (classification system followed by classification symbols)
 H04L G06Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
 EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2015/256499 A1 (KUMAR ANKESH [US] ET AL) 10 September 2015 (2015-09-10) paragraphs: 0130, 0178-0186 abstract; figures 23, 24 -----	1-10
A	US 8 832 205 B2 (LEXTINE SOFTWARE LLC [US]) 9 September 2014 (2014-09-09) column 6, line 58 - column 7, line 7 -----	2
A	US 8 751 582 B1 (BEHFOROZ REZA [US] ET AL) 10 June 2014 (2014-06-10) column 10, line 46 - column 11, line 8 column 11, line 24 - column 11, line 35 -----	4,5

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier application or patent but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "&" document member of the same patent family

Date of the actual completion of the international search
 15 February 2019

Date of mailing of the international search report
 26/02/2019

Name and mailing address of the ISA/
 European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040,
 Fax: (+31-70) 340-3016

Authorized officer
 Florou, Rafaela

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2018/062376

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015256499	A1	10-09-2015	NONE
US 8832205	B2	09-09-2014	NONE
US 8751582	B1	10-06-2014	NONE