



(51) International Patent Classification:
H04L 29/06 (2006.01)

(21) International Application Number:
PCT/CN2019/073032

(22) International Filing Date:
24 January 2019 (24.01.2019)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: CITRIX SYSTEMS, INC. [US/US]; 851 West Cypress Creek Road, Fort Lauderdale, Florida 33309 (US).

(72) Inventors: CHU, Xiaolu; c/o Citrix Nanjing, C3 Building, No. 19 Suyuan Avenue, Jiangning District, Nanjing, Jiangsu 211100 (CN). LI, Dai; c/o Citrix Nanjing, C3 Building, No. 19 Suyuan Avenue, Jiangning District, Nanjing, Jiangsu 211100 (CN).

(74) Agent: AFD CHINA INTELLECTUAL PROPERTY LAW OFFICE; Suite B 1601A, 8 Xue Qing Rd., Haidian, Beijing 100192 (CN).

(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.

(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).

Published:
— with international search report (Art. 21(3))

(54) Title: OPTIMIZED NETWORK SELECTION

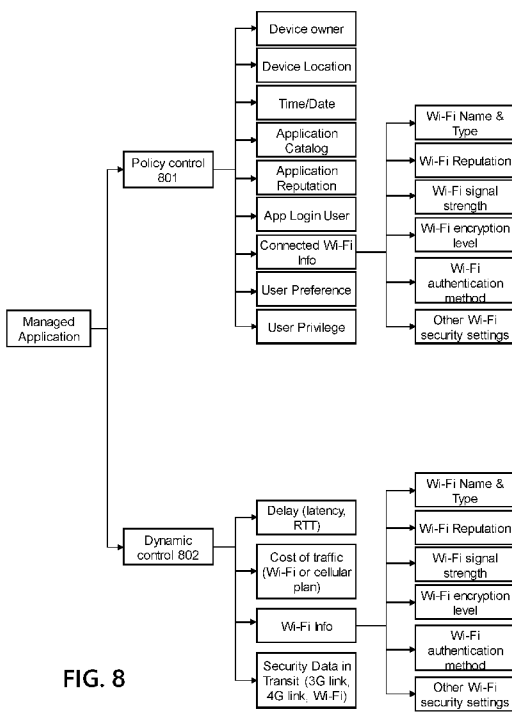


FIG. 8

(57) Abstract: Methods and systems are describe herein for optimized selection of wireless communications networks when multiple wireless communications networks are available to or for selection by a wireless communications device. A wireless communications device may select an optimal network on a per-application and/or per-transmission basis based on one or more policies defined and managed by the device and/or based on dynamic selection of a wireless network based on one or more probed network characteristics (e. g., latency, cost of traffic, data security, etc.). When a state of the device satisfies conditions specified by the policies managed and enforced on the device (e. g., in an enterprise mobility management system), then the wireless network may be selected a defined by the applicable policy. However, when the state of the device does not match an existing policy, then wireless network selection may be based the dynamic probed of the various networks to determine a preferred wireless network.



OPTIMIZED NETWORK SELECTION

FIELD

[0001] Aspects described herein generally relate to telecommunications and networking. More specifically, one or more aspects herein are directed to optimized selection of a communication network when multiple communications networks are available for communicating data between multiple devices and/or services.

BACKGROUND

[0002] Wireless communication devices often include one or more of cellular data, 3G cellular data, 4G cellular data, LTE cellular data, and/or IEEE 802.11 wireless compliant communications capabilities (e.g., 802.11n, Wi-Fi 5, and the like). However, once a device selects a network, all data communications are sent over that network, regardless of data type, service, security, etc.

SUMMARY

[0003] The following presents a simplified summary of various aspects described herein. This summary is not an extensive overview, and is not intended to identify required or critical elements or to delineate the scope of the claims. The following summary merely presents some concepts in a simplified form as an introductory prelude to the more detailed description provided below.

[0004] To overcome limitations in the prior art described above, and to overcome other limitations that will be apparent upon reading and understanding the present specification, aspects described herein are directed towards optimized selection of communications networks based on managed policies and/or dynamic testing of network characteristics.

[0005] According to an illustrative aspect, a wireless communications device may receive, from a first application executing on the wireless communications device, a request to transmit first data to a first recipient. The wireless communications device may receive, from a second application executing on the wireless communications device, a request to transmit second data to a second recipient. Upon determining that a plurality of wireless communications channels are available through which the wireless communications device can transmit the first data, the wireless communication device may select, when a state of the wireless communications device satisfies an existing policy established by a policy engine executing on the wireless communications device, a wireless communication channel

identified by the satisfied policy. However, when the state of the wireless communications device does not satisfy any existing policy established by the policy engine, the wireless communication channel may be selected based on a dynamic probe of the plurality of available wireless communications channels. The wireless communication device may then send the first data to the first recipient over a first wireless communications channel selected from the plurality of available wireless communications channels based on a first state of the wireless communications device satisfying an existing policy, where the first state is associated with the first application, and may send the second data to the second recipient over a second wireless communications channel selected from the plurality of available wireless communications channels based on the dynamic probe of the plurality of available wireless communications channels resulting from a second state of the wireless communications device not satisfying any existing policies.

[0006] In some aspects, the policy engine may store a plurality of policies, each identifying a set of one or more wireless communications channels permitted when that policy is satisfied. Each policy may be based on a plurality of variables such as device owner, device location, date, time, application name, application catalog, application reputation, connected Wi-Fi info, user login, and user privileges. Connected Wi-Fi info may be further include and be based on Wi-Fi name, Wi-Fi type, Wi-Fi reputation, Wi-Fi encryption level, and Wi-Fi authentication method.

[0007] In various aspects, one or more of the plurality of wireless communications channels may include cellular communications and/or IEEE 802.11-compliant channels.

[0008] In some aspects the dynamic probing includes determining, for each of the available wireless communications channels, delay, cost of traffic, and security, and further includes, for each available Wi-Fi communications channel, Wi-Fi name, Wi-Fi type, Wi-Fi reputation, Wi-Fi encryption level, and Wi-Fi authentication method.

[0009] Aspects may include methods, systems, wireless communications devices, and/or computer readable media storing computer executable instructions for performing the method of configuring a device or system to perform as described herein. These and additional aspects will be appreciated with the benefit of the disclosures discussed in further detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] A more complete understanding of aspects described herein and the advantages thereof may be acquired by referring to the following description in consideration of the accompanying drawings, in which like reference numbers indicate like features, and wherein:

[0011] FIG. 1 depicts an illustrative computer system architecture that may be used in accordance with one or more illustrative aspects described herein.

[0012] FIG. 2 depicts an illustrative remote-access system architecture that may be used in accordance with one or more illustrative aspects described herein.

[0013] FIG. 3 depicts an illustrative virtualized (hypervisor) system architecture that may be used in accordance with one or more illustrative aspects described herein.

[0014] FIG. 4 depicts an illustrative cloud-based system architecture that may be used in accordance with one or more illustrative aspects described herein.

[0015] FIG. 5 depicts an illustrative enterprise mobility management system.

[0016] FIG. 6 depicts another illustrative enterprise mobility management system.

[0017] FIG. 7 depicts an illustrative workflow of communication network access between a communication device and an application server that may be used in accordance with one or more illustrative aspects described herein.

[0018] FIG. 8 depicts an illustrative network selection architecture that may be used in accordance with one or more illustrative aspects described herein.

[0019] FIG. 9 depicts an exemplary user interface of a communication device that may be used in accordance with one or more illustrative aspects described herein.

[0020] FIG. 10 depicts an illustrative network access policy control architecture that may be used in accordance with one or more illustrative aspects described herein.

[0021] FIG. 11 depicts an network access dynamic control architecture that may be used in accordance with one or more illustrative aspects described herein.

[0022] FIG. 12 depicts a flowchart that illustrates a method for network selection for a communication device that may be used in accordance with one or more illustrative aspects described herein.

DETAILED DESCRIPTION

[0023] In the following description of the various embodiments, reference is made to the accompanying drawings identified above and which form a part hereof, and in which is shown by way of illustration various embodiments in which aspects described herein may be practiced. It is to be understood that other embodiments may be utilized and structural and functional modifications may be made without departing from the scope described herein. Various aspects are capable of other embodiments and of being practiced or being carried out in various different ways.

[0024] As a general introduction to the subject matter described in more detail below, aspects described herein are directed towards selecting an optimized communication network for a communication device. A managed application may perform an optimized communication network selection process that determines which communication network is to be selected for building a connection with a remote computing device for sharing data based on policy and conditional evaluation. In this way, the managed application may ensure with the intention of mitigating wireless communication network such as, Wi-Fi, with maximum security and also enhancing user experience within a software application. As a result, the user of the communication device may advantageously utilize the communication device for securely transmitting data from software applications over an optimized communication network.

[0025] It is to be understood that the phraseology and terminology used herein are for the purpose of description and should not be regarded as limiting. Rather, the phrases and terms used herein are to be given their broadest interpretation and meaning. The use of “including” and “comprising” and variations thereof is meant to encompass the items listed thereafter and equivalents thereof as well as additional items and equivalents thereof. The use of the terms “mounted,” “connected,” “coupled,” “positioned,” “engaged” and similar terms, is meant to include both direct and indirect mounting, connecting, coupling, positioning and engaging.

[0026] COMPUTING ARCHITECTURE

[0027] Computer software, hardware, and networks may be utilized in a variety of different system environments, including standalone, networked, remote-access (also known as remote desktop), virtualized, and/or cloud-based environments, among others. FIG. 1 illustrates one example of a system architecture and data processing device that may be used to implement one or more illustrative aspects described herein in a standalone and/or

networked environment. Various network nodes 103, 105, 107, and 109 may be interconnected via a wide area network (WAN) 101, such as the Internet. Other networks may also or alternatively be used, including private intranets, corporate networks, local area networks (LAN), metropolitan area networks (MAN), wireless networks, personal networks (PAN), and the like. Network 101 is for illustration purposes and may be replaced with fewer or additional computer networks. A local area network 133 may have one or more of any known LAN topology and may use one or more of a variety of different protocols, such as Ethernet. Devices 103, 105, 107, and 109 and other devices (not shown) may be connected to one or more of the networks via twisted pair wires, coaxial cable, fiber optics, radio waves, or other communication media.

[0028] The term “network” as used herein and depicted in the drawings refers not only to systems in which remote storage devices are coupled together via one or more communication paths, but also to stand-alone devices that may be coupled, from time to time, to such systems that have storage capability. Consequently, the term “network” includes not only a “physical network” but also a “content network,” which is comprised of the data—attributable to a single entity—which resides across all physical networks.

[0029] The components may include data server 103, web server 105, and client computers 107, 109. Data server 103 provides overall access, control and administration of databases and control software for performing one or more illustrative aspects describe herein. Data server 103 may be connected to web server 105 through which users interact with and obtain data as requested. Alternatively, data server 103 may act as a web server itself and be directly connected to the Internet. Data server 103 may be connected to web server 105 through the local area network 133, the wide area network 101 (e.g., the Internet), via direct or indirect connection, or via some other network. Users may interact with the data server 103 using remote computers 107, 109, e.g., using a web browser to connect to the data server 103 via one or more externally exposed web sites hosted by web server 105. Client computers 107, 109 may be used in concert with data server 103 to access data stored therein, or may be used for other purposes. For example, from client device 107 a user may access web server 105 using an Internet browser, as is known in the art, or by executing a software application that communicates with web server 105 and/or data server 103 over a computer network (such as the Internet).

[0030] Servers and applications may be combined on the same physical machines, and retain separate virtual or logical addresses, or may reside on separate physical machines. FIG.

1 illustrates just one example of a network architecture that may be used, and those of skill in the art will appreciate that the specific network architecture and data processing devices used may vary, and are secondary to the functionality that they provide, as further described herein. For example, services provided by web server 105 and data server 103 may be combined on a single server.

[0031] Each component 103, 105, 107, 109 may be any type of known computer, server, or data processing device. Data server 103, e.g., may include a processor 111 controlling overall operation of the data server 103. Data server 103 may further include random access memory (RAM) 113, read only memory (ROM) 115, network interface 117, input/output interfaces 119 (e.g., keyboard, mouse, display, printer, etc.), and memory 121. Input/output (I/O) 119 may include a variety of interface units and drives for reading, writing, displaying, and/or printing data or files. Memory 121 may further store operating system software 123 for controlling overall operation of the data processing device 103, control logic 125 for instructing data server 103 to perform aspects described herein, and other application software 127 providing secondary, support, and/or other functionality which may or might not be used in conjunction with aspects described herein. The control logic 125 may also be referred to herein as the data server software 125. Functionality of the data server software 125 may refer to operations or decisions made automatically based on rules coded into the control logic 125, made manually by a user providing input into the system, and/or a combination of automatic processing based on user input (e.g., queries, data updates, etc.).

[0032] Memory 121 may also store data used in performance of one or more aspects described herein, including a first database 129 and a second database 131. In some embodiments, the first database 129 may include the second database 131 (e.g., as a separate table, report, etc.). That is, the information can be stored in a single database, or separated into different logical, virtual, or physical databases, depending on system design. Devices 105, 107, and 109 may have similar or different architecture as described with respect to device 103. Those of skill in the art will appreciate that the functionality of data processing device 103 (or device 105, 107, or 109) as described herein may be spread across multiple data processing devices, for example, to distribute processing load across multiple computers, to segregate transactions based on geographic location, user access level, quality of service (QoS), etc.

[0033] One or more aspects may be embodied in computer-usable or readable data and/or computer-executable instructions, such as in one or more program modules, executed by one

or more computers or other devices as described herein. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types when executed by a processor in a computer or other device. The modules may be written in a source code programming language that is subsequently compiled for execution, or may be written in a scripting language such as (but not limited to) HyperText Markup Language (HTML) or Extensible Markup Language (XML). The computer executable instructions may be stored on a computer readable medium such as a nonvolatile storage device. Any suitable computer readable storage media may be utilized, including hard disks, CD-ROMs, optical storage devices, magnetic storage devices, and/or any combination thereof. In addition, various transmission (non-storage) media representing data or events as described herein may be transferred between a source and a destination in the form of electromagnetic waves traveling through signal-conducting media such as metal wires, optical fibers, and/or wireless transmission media (e.g., air and/or space). Various aspects described herein may be embodied as a method, a data processing system, or a computer program product. Therefore, various functionalities may be embodied in whole or in part in software, firmware, and/or hardware or hardware equivalents such as integrated circuits, field programmable gate arrays (FPGA), and the like. Particular data structures may be used to more effectively implement one or more aspects described herein, and such data structures are contemplated within the scope of computer executable instructions and computer-usable data described herein.

[0034] With further reference to FIG. 2, one or more aspects described herein may be implemented in a remote-access environment. FIG. 2 depicts an example system architecture including a computing device 201 in an illustrative computing environment 200 that may be used according to one or more illustrative aspects described herein. Computing device 201 may be used as a server 206a in a single-server or multi-server desktop virtualization system (e.g., a remote access or cloud system) and can be configured to provide virtual machines for client access devices. The computing device 201 may have a processor 203 for controlling overall operation of the device 201 and its associated components, including RAM 205, ROM 207, Input/Output (I/O) module 209, and memory 215.

[0035] I/O module 209 may include a mouse, keypad, touch screen, scanner, optical reader, and/or stylus (or other input device(s)) through which a user of computing device 201 may provide input, and may also include one or more of a speaker for providing audio output and one or more of a video display device for providing textual, audiovisual, and/or graphical

output. Software may be stored within memory 215 and/or other storage to provide instructions to processor 203 for configuring computing device 201 into a special purpose computing device in order to perform various functions as described herein. For example, memory 215 may store software used by the computing device 201, such as an operating system 217, application programs 219, and an associated database 221.

[0036] Computing device 201 may operate in a networked environment supporting connections to one or more remote computers, such as terminals 240 (also referred to as client devices). The terminals 240 may be personal computers, mobile devices, laptop computers, tablets, or servers that include many or all of the elements described above with respect to the computing device 103 or 201. The network connections depicted in FIG. 2 include a local area network (LAN) 225 and a wide area network (WAN) 229, but may also include other networks. When used in a LAN networking environment, computing device 201 may be connected to the LAN 225 through a network interface or adapter 223. When used in a WAN networking environment, computing device 201 may include a modem or other wide area network interface 227 for establishing communications over the WAN 229, such as computer network 230 (e.g., the Internet). It will be appreciated that the network connections shown are illustrative and other means of establishing a communications link between the computers may be used. Computing device 201 and/or terminals 240 may also be mobile terminals (e.g., mobile phones, smartphones, personal digital assistants (PDAs), notebooks, etc.) including various other components, such as a battery, speaker, and antennas (not shown).

[0037] Aspects described herein may also be operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of other computing systems, environments, and/or configurations that may be suitable for use with aspects described herein include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network personal computers (PCs), minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

[0038] As shown in FIG. 2, one or more client devices 240 may be in communication with one or more servers 206a-206n (generally referred to herein as “server(s) 206”). In one embodiment, the computing environment 200 may include a network appliance installed between the server(s) 206 and client machine(s) 240. The network appliance may manage

client/server connections, and in some cases can load balance client connections amongst a plurality of backend servers 206.

[0039] The client machine(s) 240 may in some embodiments be referred to as a single client machine 240 or a single group of client machines 240, while server(s) 206 may be referred to as a single server 206 or a single group of servers 206. In one embodiment a single client machine 240 communicates with more than one server 206, while in another embodiment a single server 206 communicates with more than one client machine 240. In yet another embodiment, a single client machine 240 communicates with a single server 206.

[0040] A client machine 240 can, in some embodiments, be referenced by any one of the following non-exhaustive terms: client machine(s); client(s); client computer(s); client device(s); client computing device(s); local machine; remote machine; client node(s); endpoint(s); or endpoint node(s). The server 206, in some embodiments, may be referenced by any one of the following non-exhaustive terms: server(s), local machine; remote machine; server farm(s), or host computing device(s).

[0041] In one embodiment, the client machine 240 may be a virtual machine. The virtual machine may be any virtual machine, while in some embodiments the virtual machine may be any virtual machine managed by a Type 1 or Type 2 hypervisor, for example, a hypervisor developed by Citrix Systems, IBM, VMware, or any other hypervisor. In some aspects, the virtual machine may be managed by a hypervisor, while in other aspects the virtual machine may be managed by a hypervisor executing on a server 206 or a hypervisor executing on a client 240.

[0042] Some embodiments include a client device 240 that displays application output generated by an application remotely executing on a server 206 or other remotely located machine. In these embodiments, the client device 240 may execute a virtual machine receiver program or application to display the output in an application window, a browser, or other output window. In one example, the application is a desktop, while in other examples the application is an application that generates or presents a desktop. A desktop may include a graphical shell providing a user interface for an instance of an operating system in which local and/or remote applications can be integrated. Applications, as used herein, are programs that execute after an instance of an operating system (and, optionally, also the desktop) has been loaded.

[0043] The server 206, in some embodiments, uses a remote presentation protocol or other program to send data to a thin-client or remote-display application executing on the client to present display output generated by an application executing on the server 206. The thin-client or remote-display protocol can be any one of the following non-exhaustive list of protocols: the Independent Computing Architecture (ICA) protocol developed by Citrix Systems, Inc. of Ft. Lauderdale, Florida; or the Remote Desktop Protocol (RDP) manufactured by the Microsoft Corporation of Redmond, Washington.

[0044] A remote computing environment may include more than one server 206a-206n such that the servers 206a-206n are logically grouped together into a server farm 206, for example, in a cloud computing environment. The server farm 206 may include servers 206 that are geographically dispersed while logically grouped together, or servers 206 that are located proximate to each other while logically grouped together. Geographically dispersed servers 206a-206n within a server farm 206 can, in some embodiments, communicate using a WAN (wide), MAN (metropolitan), or LAN (local), where different geographic regions can be characterized as: different continents; different regions of a continent; different countries; different states; different cities; different campuses; different rooms; or any combination of the preceding geographical locations. In some embodiments the server farm 206 may be administered as a single entity, while in other embodiments the server farm 206 can include multiple server farms.

[0045] In some embodiments, a server farm may include servers 206 that execute a substantially similar type of operating system platform (e.g., WINDOWS, UNIX, LINUX, iOS, ANDROID, SYMBIAN, etc.) In other embodiments, server farm 206 may include a first group of one or more servers that execute a first type of operating system platform, and a second group of one or more servers that execute a second type of operating system platform.

[0046] Server 206 may be configured as any type of server, as needed, e.g., a file server, an application server, a web server, a proxy server, an appliance, a network appliance, a gateway, an application gateway, a gateway server, a virtualization server, a deployment server, a Secure Sockets Layer (SSL) VPN server, a firewall, a web server, an application server or as a master application server, a server executing an active directory, or a server executing an application acceleration program that provides firewall functionality, application functionality, or load balancing functionality. Other server types may also be used.

[0047] Some embodiments include a first server 206a that receives requests from a client machine 240, forwards the request to a second server 206b (not shown), and responds to the request generated by the client machine 240 with a response from the second server 206b (not shown.) First server 206a may acquire an enumeration of applications available to the client machine 240 as well as address information associated with an application server 206 hosting an application identified within the enumeration of applications. First server 206a can then present a response to the client's request using a web interface, and communicate directly with the client 240 to provide the client 240 with access to an identified application. One or more clients 240 and/or one or more servers 206 may transmit data over network 230, e.g., network 101.

[0048] FIG. 3 shows a high-level architecture of an illustrative desktop virtualization system. As shown, the desktop virtualization system may be single-server or multi-server system, or cloud system, including at least one virtualization server 301 configured to provide virtual desktops and/or virtual applications to one or more client access devices 240. As used herein, a desktop refers to a graphical environment or space in which one or more applications may be hosted and/or executed. A desktop may include a graphical shell providing a user interface for an instance of an operating system in which local and/or remote applications can be integrated. Applications may include programs that execute after an instance of an operating system (and, optionally, also the desktop) has been loaded. Each instance of the operating system may be physical (e.g., one operating system per device) or virtual (e.g., many instances of an OS running on a single device). Each application may be executed on a local device, or executed on a remotely located device (e.g., remoted).

[0049] A computer device 301 may be configured as a virtualization server in a virtualization environment, for example, a single-server, multi-server, or cloud computing environment. Virtualization server 301 illustrated in FIG. 3 can be deployed as and/or implemented by one or more embodiments of the server 206 illustrated in FIG. 2 or by other known computing devices. Included in virtualization server 301 is a hardware layer that can include one or more physical disks 304, one or more physical devices 306, one or more physical processors 308, and one or more physical memories 316. In some embodiments, firmware 312 can be stored within a memory element in the physical memory 316 and can be executed by one or more of the physical processors 308. Virtualization server 301 may further include an operating system 314 that may be stored in a memory element in the physical memory 316 and executed by one or more of the physical processors 308. Still

further, a hypervisor 302 may be stored in a memory element in the physical memory 316 and can be executed by one or more of the physical processors 308.

[0050] Executing on one or more of the physical processors 308 may be one or more virtual machines 332A-C (generally 332). Each virtual machine 332 may have a virtual disk 326A-C and a virtual processor 328A-C. In some embodiments, a first virtual machine 332A may execute, using a virtual processor 328A, a control program 320 that includes a tools stack 324. Control program 320 may be referred to as a control virtual machine, Dom0, Domain 0, or other virtual machine used for system administration and/or control. In some embodiments, one or more virtual machines 332B-C can execute, using a virtual processor 328B-C, a guest operating system 330A-B.

[0051] Virtualization server 301 may include a hardware layer 310 with one or more pieces of hardware that communicate with the virtualization server 301. In some embodiments, the hardware layer 310 can include one or more physical disks 304, one or more physical devices 306, one or more physical processors 308, and one or more physical memory 316. Physical components 304, 306, 308, and 316 may include, for example, any of the components described above. Physical devices 306 may include, for example, a network interface card, a video card, a keyboard, a mouse, an input device, a monitor, a display device, speakers, an optical drive, a storage device, a universal serial bus connection, a printer, a scanner, a network element (e.g., router, firewall, network address translator, load balancer, virtual private network (VPN) gateway, Dynamic Host Configuration Protocol (DHCP) router, etc.), or any device connected to or communicating with virtualization server 301. Physical memory 316 in the hardware layer 310 may include any type of memory. Physical memory 316 may store data, and in some embodiments may store one or more programs, or set of executable instructions. FIG. 3 illustrates an embodiment where firmware 312 is stored within the physical memory 316 of virtualization server 301. Programs or executable instructions stored in the physical memory 316 can be executed by the one or more processors 308 of virtualization server 301.

[0052] Virtualization server 301 may also include a hypervisor 302. In some embodiments, hypervisor 302 may be a program executed by processors 308 on virtualization server 301 to create and manage any number of virtual machines 332. Hypervisor 302 may be referred to as a virtual machine monitor, or platform virtualization software. In some embodiments, hypervisor 302 can be any combination of executable instructions and hardware that monitors virtual machines executing on a computing machine. Hypervisor 302

may be Type 2 hypervisor, where the hypervisor executes within an operating system 314 executing on the virtualization server 301. Virtual machines may then execute at a level above the hypervisor 302. In some embodiments, the Type 2 hypervisor may execute within the context of a user's operating system such that the Type 2 hypervisor interacts with the user's operating system. In other embodiments, one or more virtualization servers 301 in a virtualization environment may instead include a Type 1 hypervisor (not shown). A Type 1 hypervisor may execute on the virtualization server 301 by directly accessing the hardware and resources within the hardware layer 310. That is, while a Type 2 hypervisor 302 accesses system resources through a host operating system 314, as shown, a Type 1 hypervisor may directly access all system resources without the host operating system 314. A Type 1 hypervisor may execute directly on one or more physical processors 308 of virtualization server 301, and may include program data stored in the physical memory 316.

[0053] Hypervisor 302, in some embodiments, can provide virtual resources to operating systems 330 or control programs 320 executing on virtual machines 332 in any manner that simulates the operating systems 330 or control programs 320 having direct access to system resources. System resources can include, but are not limited to, physical devices 306, physical disks 304, physical processors 308, physical memory 316, and any other component included in hardware layer 310 of the virtualization server 301. Hypervisor 302 may be used to emulate virtual hardware, partition physical hardware, virtualize physical hardware, and/or execute virtual machines that provide access to computing environments. In still other embodiments, hypervisor 302 may control processor scheduling and memory partitioning for a virtual machine 332 executing on virtualization server 301. Hypervisor 302 may include those manufactured by VMWare, Inc., of Palo Alto, California; the XENPROJECT hypervisor, an open source product whose development is overseen by the open source XenProject.org community; HyperV, VirtualServer or virtual PC hypervisors provided by Microsoft, or others. In some embodiments, virtualization server 301 may execute a hypervisor 302 that creates a virtual machine platform on which guest operating systems may execute. In these embodiments, the virtualization server 301 may be referred to as a host server. An example of such a virtualization server is the XENSERVEN provided by Citrix Systems, Inc., of Fort Lauderdale, FL.

[0054] Hypervisor 302 may create one or more virtual machines 332B-C (generally 332) in which guest operating systems 330 execute. In some embodiments, hypervisor 302 may load a virtual machine image to create a virtual machine 332. In other embodiments, the

hypervisor 302 may execute a guest operating system 330 within virtual machine 332. In still other embodiments, virtual machine 332 may execute guest operating system 330.

[0055] In addition to creating virtual machines 332, hypervisor 302 may control the execution of at least one virtual machine 332. In other embodiments, hypervisor 302 may present at least one virtual machine 332 with an abstraction of at least one hardware resource provided by the virtualization server 301 (e.g., any hardware resource available within the hardware layer 310). In other embodiments, hypervisor 302 may control the manner in which virtual machines 332 access physical processors 308 available in virtualization server 301. Controlling access to physical processors 308 may include determining whether a virtual machine 332 should have access to a processor 308, and how physical processor capabilities are presented to the virtual machine 332.

[0056] As shown in FIG. 3, virtualization server 301 may host or execute one or more virtual machines 332. A virtual machine 332 is a set of executable instructions that, when executed by a processor 308, may imitate the operation of a physical computer such that the virtual machine 332 can execute programs and processes much like a physical computing device. While FIG. 3 illustrates an embodiment where a virtualization server 301 hosts three virtual machines 332, in other embodiments virtualization server 301 can host any number of virtual machines 332. Hypervisor 302, in some embodiments, may provide each virtual machine 332 with a unique virtual view of the physical hardware, memory, processor, and other system resources available to that virtual machine 332. In some embodiments, the unique virtual view can be based on one or more of virtual machine permissions, application of a policy engine to one or more virtual machine identifiers, a user accessing a virtual machine, the applications executing on a virtual machine, networks accessed by a virtual machine, or any other desired criteria. For instance, hypervisor 302 may create one or more unsecure virtual machines 332 and one or more secure virtual machines 332. Unsecure virtual machines 332 may be prevented from accessing resources, hardware, memory locations, and programs that secure virtual machines 332 may be permitted to access. In other embodiments, hypervisor 302 may provide each virtual machine 332 with a substantially similar virtual view of the physical hardware, memory, processor, and other system resources available to the virtual machines 332.

[0057] Each virtual machine 332 may include a virtual disk 326A-C (generally 326) and a virtual processor 328A-C (generally 328.) The virtual disk 326, in some embodiments, is a virtualized view of one or more physical disks 304 of the virtualization server 301, or a

portion of one or more physical disks 304 of the virtualization server 301. The virtualized view of the physical disks 304 can be generated, provided, and managed by the hypervisor 302. In some embodiments, hypervisor 302 provides each virtual machine 332 with a unique view of the physical disks 304. Thus, in these embodiments, the particular virtual disk 326 included in each virtual machine 332 can be unique when compared with the other virtual disks 326.

[0058] A virtual processor 328 can be a virtualized view of one or more physical processors 308 of the virtualization server 301. In some embodiments, the virtualized view of the physical processors 308 can be generated, provided, and managed by hypervisor 302. In some embodiments, virtual processor 328 has substantially all of the same characteristics of at least one physical processor 308. In other embodiments, virtual processor 308 provides a modified view of physical processors 308 such that at least some of the characteristics of the virtual processor 328 are different than the characteristics of the corresponding physical processor 308.

[0059] With further reference to FIG. 4, some aspects described herein may be implemented in a cloud-based environment. FIG. 4 illustrates an example of a cloud computing environment (or cloud system) 400. As seen in FIG. 4, client computers 411-414 may communicate with a cloud management server 410 to access the computing resources (e.g., host servers 403a-403b (generally referred herein as “host servers 403”), storage resources 404a-404b (generally referred herein as “storage resources 404”), and network elements 405a-405b (generally referred herein as “network resources 405”)) of the cloud system.

[0060] Management server 410 may be implemented on one or more physical servers. The management server 410 may run, for example, CLOUDPLATFORM by Citrix Systems, Inc. of Ft. Lauderdale, FL, or OPENSTACK, among others. Management server 410 may manage various computing resources, including cloud hardware and software resources, for example, host computers 403, data storage devices 404, and networking devices 405. The cloud hardware and software resources may include private and/or public components. For example, a cloud may be configured as a private cloud to be used by one or more particular customers or client computers 411-414 and/or over a private network. In other embodiments, public clouds or hybrid public-private clouds may be used by other customers over an open or hybrid networks.

[0061] Management server 410 may be configured to provide user interfaces through which cloud operators and cloud customers may interact with the cloud system 400. For example, the management server 410 may provide a set of application programming interfaces (APIs) and/or one or more cloud operator console applications (e.g., web-based or standalone applications) with user interfaces to allow cloud operators to manage the cloud resources, configure the virtualization layer, manage customer accounts, and perform other cloud administration tasks. The management server 410 also may include a set of APIs and/or one or more customer console applications with user interfaces configured to receive cloud computing requests from end users via client computers 411-414, for example, requests to create, modify, or destroy virtual machines within the cloud. Client computers 411-414 may connect to management server 410 via the Internet or some other communication network, and may request access to one or more of the computing resources managed by management server 410. In response to client requests, the management server 410 may include a resource manager configured to select and provision physical resources in the hardware layer of the cloud system based on the client requests. For example, the management server 410 and additional components of the cloud system may be configured to provision, create, and manage virtual machines and their operating environments (e.g., hypervisors, storage resources, services offered by the network elements, etc.) for customers at client computers 411-414, over a network (e.g., the Internet), providing customers with computational resources, data storage services, networking capabilities, and computer platform and application support. Cloud systems also may be configured to provide various specific services, including security systems, development environments, user interfaces, and the like.

[0062] Certain clients 411-414 may be related, for example, to different client computers creating virtual machines on behalf of the same end user, or different users affiliated with the same company or organization. In other examples, certain clients 411-414 may be unrelated, such as users affiliated with different companies or organizations. For unrelated clients, information on the virtual machines or storage of any one user may be hidden from other users.

[0063] Referring now to the physical hardware layer of a cloud computing environment, availability zones 401-402 (or zones) may refer to a collocated set of physical computing resources. Zones may be geographically separated from other zones in the overall cloud of computing resources. For example, zone 401 may be a first cloud datacenter located in California, and zone 402 may be a second cloud datacenter located in Florida. Management

server 410 may be located at one of the availability zones, or at a separate location. Each zone may include an internal network that interfaces with devices that are outside of the zone, such as the management server 410, through a gateway. End users of the cloud (e.g., clients 411-414) might or might not be aware of the distinctions between zones. For example, an end user may request the creation of a virtual machine having a specified amount of memory, processing power, and network capabilities. The management server 410 may respond to the user's request and may allocate the resources to create the virtual machine without the user knowing whether the virtual machine was created using resources from zone 401 or zone 402. In other examples, the cloud system may allow end users to request that virtual machines (or other cloud resources) are allocated in a specific zone or on specific resources 403-405 within a zone.

[0064] In this example, each zone 401-402 may include an arrangement of various physical hardware components (or computing resources) 403-405, for example, physical hosting resources (or processing resources), physical network resources, physical storage resources, switches, and additional hardware resources that may be used to provide cloud computing services to customers. The physical hosting resources in a cloud zone 401-402 may include one or more computer servers 403, such as the virtualization servers 301 described above, which may be configured to create and host virtual machine instances. The physical network resources in a cloud zone 401 or 402 may include one or more network elements 405 (e.g., network service providers) comprising hardware and/or software configured to provide a network service to cloud customers, such as firewalls, network address translators, load balancers, virtual private network (VPN) gateways, Dynamic Host Configuration Protocol (DHCP) routers, and the like. The storage resources in the cloud zone 401-402 may include storage disks (e.g., solid state drives (SSDs), magnetic hard disks, etc.) and other storage devices.

[0065] The example cloud computing environment shown in FIG. 4 also may include a virtualization layer (e.g., as shown in FIGS. 1-3) with additional hardware and/or software resources configured to create and manage virtual machines and provide other services to customers using the physical resources in the cloud. The virtualization layer may include hypervisors, as described above in FIG. 3, along with other components to provide network virtualizations, storage virtualizations, etc. The virtualization layer may be as a separate layer from the physical resource layer, or may share some or all of the same hardware and/or software resources with the physical resource layer. For example, the virtualization layer may

include a hypervisor installed in each of the virtualization servers 403 with the physical computing resources. Known cloud systems may alternatively be used, e.g., WINDOWS AZURE (Microsoft Corporation of Redmond Washington), AMAZON EC2 (Amazon.com Inc. of Seattle, Washington), IBM BLUE CLOUD (IBM Corporation of Armonk, New York), or others.

[0066] ENTERPRISE MOBILITY MANAGEMENT ARCHITECTURE

[0067] FIG. 5 represents an enterprise mobility technical architecture 500 for use in a “Bring Your Own Device” (BYOD) environment. The architecture enables a user of a mobile device 502 to both access enterprise or personal resources from a mobile device 502 and use the mobile device 502 for personal use. The user may access such enterprise resources 504 or enterprise services 508 using a mobile device 502 that is purchased by the user or a mobile device 502 that is provided by the enterprise to the user. The user may utilize the mobile device 502 for business use only or for business and personal use. The mobile device 502 may run an iOS operating system, an Android operating system, or the like. The enterprise may choose to implement policies to manage the mobile device 502. The policies may be implemented through a firewall or gateway in such a way that the mobile device 502 may be identified, secured or security verified, and provided selective or full access to the enterprise resources (e.g., 504 and 508.) The policies may be mobile device management policies, mobile application management policies, mobile data management policies, or some combination of mobile device, application, and data management policies. A mobile device 502 that is managed through the application of mobile device management policies may be referred to as an enrolled device.

[0068] In some embodiments, the operating system of the mobile device 502 may be separated into a managed partition 510 and an unmanaged partition 512. The managed partition 510 may have policies applied to it to secure the applications running on and data stored in the managed partition 510. The applications running on the managed partition 510 may be secure applications. In other embodiments, all applications may execute in accordance with a set of one or more policy files received separate from the application, and which define one or more security parameters, features, resource restrictions, and/or other access controls that are enforced by the mobile device management system when that application is executing on the mobile device 502. By operating in accordance with their respective policy file(s), each application may be allowed or restricted from communications with one or more other applications and/or resources, thereby creating a virtual partition.

Thus, as used herein, a partition may refer to a physically partitioned portion of memory (physical partition), a logically partitioned portion of memory (logical partition), and/or a virtual partition created as a result of enforcement of one or more policies and/or policy files across multiple applications as described herein (virtual partition). Stated differently, by enforcing policies on managed applications, those applications may be restricted to only be able to communicate with other managed applications and trusted enterprise resources, thereby creating a virtual partition that is not accessible by unmanaged applications and devices.

[0069] The secure applications may be email applications, web browsing applications, software-as-a-service (SaaS) access applications, Windows Application access applications, and the like. The secure applications may be secure native applications 514, secure remote applications 522 executed by a secure application launcher 518, virtualization applications 526 executed by a secure application launcher 518, and the like. The secure native applications 514 may be wrapped by a secure application wrapper 520. The secure application wrapper 520 may include integrated policies that are executed on the mobile device 502 when the secure native application 514 is executed on the mobile device 502. The secure application wrapper 520 may include meta-data that points the secure native application 514 running on the mobile device 502 to the resources hosted at the enterprise (e.g., 504 and 508) that the secure native application 514 may require to complete the task requested upon execution of the secure native application 514. The secure remote applications 522 executed by a secure application launcher 518 may be executed within the secure application launcher 518. The virtualization applications 526 executed by a secure application launcher 518 may utilize resources on the mobile device 502, at the enterprise resources 504, and the like. The resources used on the mobile device 502 by the virtualization applications 526 executed by a secure application launcher 518 may include user interaction resources, processing resources, and the like. The user interaction resources may be used to collect and transmit keyboard input, mouse input, camera input, tactile input, audio input, visual input, gesture input, and the like. The processing resources may be used to present a user interface, process data received from the enterprise resources 504, and the like. The resources used at the enterprise resources 504 by the virtualization applications 526 executed by a secure application launcher 518 may include user interface generation resources, processing resources, and the like. The user interface generation resources may be used to assemble a user interface, modify a user interface, refresh a user interface, and the like. The

processing resources may be used to create information, read information, update information, delete information, and the like. For example, the virtualization application 526 may record user interactions associated with a graphical user interface (GUI) and communicate them to a server application where the server application will use the user interaction data as an input to the application operating on the server. In such an arrangement, an enterprise may elect to maintain the application on the server side as well as data, files, etc. associated with the application. While an enterprise may elect to “mobilize” some applications in accordance with the principles herein by securing them for deployment on the mobile device 502, this arrangement may also be elected for certain applications. For example, while some applications may be secured for use on the mobile device 502, others might not be prepared or appropriate for deployment on the mobile device 502 so the enterprise may elect to provide the mobile user access to the unprepared applications through virtualization techniques. As another example, the enterprise may have large complex applications with large and complex data sets (e.g., material resource planning applications) where it would be very difficult, or otherwise undesirable, to customize the application for the mobile device 502 so the enterprise may elect to provide access to the application through virtualization techniques. As yet another example, the enterprise may have an application that maintains highly secured data (e.g., human resources data, customer data, engineering data) that may be deemed by the enterprise as too sensitive for even the secured mobile environment so the enterprise may elect to use virtualization techniques to permit mobile access to such applications and data. An enterprise may elect to provide both fully secured and fully functional applications on the mobile device 502 as well as a virtualization application 526 to allow access to applications that are deemed more properly operated on the server side. In an embodiment, the virtualization application 526 may store some data, files, etc. on the mobile device 502 in one of the secure storage locations. An enterprise, for example, may elect to allow certain information to be stored on the mobile device 502 while not permitting other information.

[0070] In connection with the virtualization application 526, as described herein, the mobile device 502 may have a virtualization application 526 that is designed to present GUIs and then record user interactions with the GUI. The virtualization application 526 may communicate the user interactions to the server side to be used by the server side application as user interactions with the application. In response, the application on the server side may transmit back to the mobile device 502 a new GUI. For example, the new GUI may be a static

page, a dynamic page, an animation, or the like, thereby providing access to remotely located resources.

[0071] The secure applications 514 may access data stored in a secure data container 528 in the managed partition 510 of the mobile device 502. The data secured in the secure data container may be accessed by the secure native applications 514, secure remote applications 522 executed by a secure application launcher 518, virtualization applications 526 executed by a secure application launcher 518, and the like. The data stored in the secure data container 528 may include files, databases, and the like. The data stored in the secure data container 528 may include data restricted to a specific secure application 530, shared among secure applications 532, and the like. Data restricted to a secure application may include secure general data 534 and highly secure data 538. Secure general data may use a strong form of encryption such as Advanced Encryption Standard (AES) 128-bit encryption or the like, while highly secure data 538 may use a very strong form of encryption such as AES 256-bit encryption. Data stored in the secure data container 528 may be deleted from the mobile device 502 upon receipt of a command from the device manager 524. The secure applications (e.g., 514, 522, and 526) may have a dual-mode option 540. The dual mode option 540 may present the user with an option to operate the secured application in an unsecured or unmanaged mode. In an unsecured or unmanaged mode, the secure applications may access data stored in an unsecured data container 542 on the unmanaged partition 512 of the mobile device 502. The data stored in an unsecured data container may be personal data 544. The data stored in an unsecured data container 542 may also be accessed by unsecured applications 546 that are running on the unmanaged partition 512 of the mobile device 502. The data stored in an unsecured data container 542 may remain on the mobile device 502 when the data stored in the secure data container 528 is deleted from the mobile device 502. An enterprise may want to delete from the mobile device 502 selected or all data, files, and/or applications owned, licensed or controlled by the enterprise (enterprise data) while leaving or otherwise preserving personal data, files, and/or applications owned, licensed or controlled by the user (personal data). This operation may be referred to as a selective wipe. With the enterprise and personal data arranged in accordance to the aspects described herein, an enterprise may perform a selective wipe.

[0072] The mobile device 502 may connect to enterprise resources 504 and enterprise services 508 at an enterprise, to the public Internet 548, and the like. The mobile device 502 may connect to enterprise resources 504 and enterprise services 508 through virtual private

network connections. The virtual private network connections, also referred to as microVPN or application-specific VPN, may be specific to particular applications (as illustrated by microVPNs 550, particular devices, particular secured areas on the mobile device (as illustrated by O/S VPN 552), and the like. For example, each of the wrapped applications in the secured area of the mobile device 502 may access enterprise resources through an application specific VPN such that access to the VPN would be granted based on attributes associated with the application, possibly in conjunction with user or device attribute information. The virtual private network connections may carry Microsoft Exchange traffic, Microsoft Active Directory traffic, HyperText Transfer Protocol (HTTP) traffic, HyperText Transfer Protocol Secure (HTTPS) traffic, application management traffic, and the like. The virtual private network connections may support and enable single-sign-on authentication processes 554. The single-sign-on processes may allow a user to provide a single set of authentication credentials, which are then verified by an authentication service 558. The authentication service 558 may then grant to the user access to multiple enterprise resources 504, without requiring the user to provide authentication credentials to each individual enterprise resource 504.

[0073] The virtual private network connections may be established and managed by an access gateway 560. The access gateway 560 may include performance enhancement features that manage, accelerate, and improve the delivery of enterprise resources 504 to the mobile device 502. The access gateway 560 may also re-route traffic from the mobile device 502 to the public Internet 548, enabling the mobile device 502 to access publicly available and unsecured applications that run on the public Internet 548. The mobile device 502 may connect to the access gateway via a transport network 562. The transport network 562 may use one or more transport protocols and may be a wired network, wireless network, cloud network, local area network, metropolitan area network, wide area network, public network, private network, and the like.

[0074] The enterprise resources 504 may include email servers, file sharing servers, SaaS applications, Web application servers, Windows application servers, and the like. Email servers may include Exchange servers, Lotus Notes servers, and the like. File sharing servers may include ShareFile servers, and the like. SaaS applications may include Salesforce, and the like. Windows application servers may include any application server that is built to provide applications that are intended to run on a local Windows operating system, and the like. The enterprise resources 504 may be premise-based resources, cloud-based resources,

and the like. The enterprise resources 504 may be accessed by the mobile device 502 directly or through the access gateway 560. The enterprise resources 504 may be accessed by the mobile device 502 via the transport network 562.

[0075] The enterprise services 508 may include authentication services 558, threat detection services 564, device manager services 524, file sharing services 568, policy manager services 570, social integration services 572, application controller services 574, and the like. Authentication services 558 may include user authentication services, device authentication services, application authentication services, data authentication services, and the like. Authentication services 558 may use certificates. The certificates may be stored on the mobile device 502, by the enterprise resources 504, and the like. The certificates stored on the mobile device 502 may be stored in an encrypted location on the mobile device 502, the certificate may be temporarily stored on the mobile device 502 for use at the time of authentication, and the like. Threat detection services 564 may include intrusion detection services, unauthorized access attempt detection services, and the like. Unauthorized access attempt detection services may include unauthorized attempts to access devices, applications, data, and the like. Device management services 524 may include configuration, provisioning, security, support, monitoring, reporting, and decommissioning services. File sharing services 568 may include file management services, file storage services, file collaboration services, and the like. Policy manager services 570 may include device policy manager services, application policy manager services, data policy manager services, and the like. Social integration services 572 may include contact integration services, collaboration services, integration with social networks such as Facebook, Twitter, and LinkedIn, and the like. Application controller services 574 may include management services, provisioning services, deployment services, assignment services, revocation services, wrapping services, and the like.

[0076] The enterprise mobility technical architecture 500 may include an application store 578. The application store 578 may include unwrapped applications 580, pre-wrapped applications 582, and the like. Applications may be populated in the application store 578 from the application controller 574. The application store 578 may be accessed by the mobile device 502 through the access gateway 560, through the public Internet 548, or the like. The application store 578 may be provided with an intuitive and easy to use user interface.

[0077] A software development kit 584 may provide a user the capability to secure applications selected by the user by wrapping the application as described previously in this

description. An application that has been wrapped using the software development kit 584 may then be made available to the mobile device 502 by populating it in the application store 578 using the application controller 574.

[0078] The enterprise mobility technical architecture 500 may include a management and analytics capability 588. The management and analytics capability 588 may provide information related to how resources are used, how often resources are used, and the like. Resources may include devices, applications, data, and the like. How resources are used may include which devices download which applications, which applications access which data, and the like. How often resources are used may include how often an application has been downloaded, how many times a specific set of data has been accessed by an application, and the like.

[0079] FIG. 6 is another illustrative enterprise mobility management system 600. Some of the components of the mobility management system 500 described above with reference to FIG. 5 have been omitted for the sake of simplicity. The architecture of the system 600 depicted in FIG. 6 is similar in many respects to the architecture of the system 500 described above with reference to FIG. 5 and may include additional features not mentioned above.

[0080] In this case, the left hand side represents an enrolled mobile device 602 with a client agent 604, which interacts with gateway server 606 (which includes Access Gateway and application controller functionality) to access various enterprise resources 608 and services 609 such as Exchange, Sharepoint, public-key infrastructure (PKI) Resources, Kerberos Resources, Certificate Issuance service, as shown on the right hand side above. Although not specifically shown, the mobile device 602 may also interact with an enterprise application store (StoreFront) for the selection and downloading of applications.

[0081] The client agent 604 acts as the UI (user interface) intermediary for Windows apps/desktops hosted in an Enterprise data center, which are accessed using the High-Definition User Experience (HDX)/ICA display remoting protocol. The client agent 604 also supports the installation and management of native applications on the mobile device 602, such as native iOS or Android applications. For example, the managed applications 610 (mail, browser, wrapped application) shown in the figure above are all native applications that execute locally on the mobile device 602. Client agent 604 and application management framework of this architecture act to provide policy driven management capabilities and features such as connectivity and SSO (single sign on) to enterprise resources/services 608.

The client agent 604 handles primary user authentication to the enterprise, normally to Access Gateway (AG) 606 with SSO to other gateway server components. The client agent 604 obtains policies from gateway server 606 to control the behavior of the managed applications 610 on the mobile device 602.

[0082] The Secure InterProcess Communication (IPC) links 612 between the native applications 610 and client agent 604 represent a management channel, which may allow a client agent to supply policies to be enforced by the application management framework 614 “wrapping” each application. The IPC channel 612 may also allow client agent 604 to supply credential and authentication information that enables connectivity and SSO to enterprise resources 608. Finally, the IPC channel 612 may allow the application management framework 614 to invoke user interface functions implemented by client agent 604, such as online and offline authentication.

[0083] Communications between the client agent 604 and gateway server 606 are essentially an extension of the management channel from the application management framework 614 wrapping each native managed application 610. The application management framework 614 may request policy information from client agent 604, which in turn may request it from gateway server 606. The application management framework 614 may request authentication, and client agent 604 may log into the gateway services part of gateway server 606 (also known as NETSCALER ACCESS GATEWAY). Client agent 604 may also call supporting services on gateway server 606, which may produce input material to derive encryption keys for the local data vaults 616, or may provide client certificates which may enable direct authentication to PKI protected resources, as more fully explained below.

[0084] In more detail, the application management framework 614 “wraps” each managed application 610. This may be incorporated via an explicit build step, or via a post-build processing step. The application management framework 614 may “pair” with client agent 604 on first launch of an application 610 to initialize the Secure IPC channel 612 and obtain the policy for that application. The application management framework 614 may enforce relevant portions of the policy that apply locally, such as the client agent login dependencies and some of the containment policies that restrict how local OS services may be used, or how they may interact with the managed application 610.

[0085] The application management framework 614 may use services provided by client agent 604 over the Secure IPC channel 612 to facilitate authentication and internal network

access. Key management for the private and shared data vaults 616 (containers) may be also managed by appropriate interactions between the managed applications 610 and client agent 604. Vaults 616 may be available only after online authentication, or may be made available after offline authentication if allowed by policy. First use of vaults 616 may require online authentication, and offline access may be limited to at most the policy refresh period before online authentication is again required.

[0086] Network access to internal resources may occur directly from individual managed applications 610 through Access Gateway 606. The application management framework 614 may be responsible for orchestrating the network access on behalf of each managed application 610. Client agent 604 may facilitate these network connections by providing suitable time limited secondary credentials obtained following online authentication. Multiple modes of network connection may be used, such as reverse web proxy connections and end-to-end VPN-style tunnels 618.

[0087] The Mail and Browser managed applications 610 have special status and may make use of facilities that might not be generally available to arbitrary wrapped applications. For example, the Mail application 610 may use a special background network access mechanism that allows it to access an Exchange server 608 over an extended period of time without requiring a full AG logon. The Browser application 610 may use multiple private data vaults 616 to segregate different kinds of data.

[0088] This architecture may support the incorporation of various other security features. For example, gateway server 606 (including its gateway services) in some cases may not need to validate active directory (AD) passwords. It can be left to the discretion of an enterprise whether an AD password may be used as an authentication factor for some users in some situations. Different authentication methods may be used if a user is online or offline (i.e., connected or not connected to a network).

[0089] Step up authentication is a feature wherein gateway server 606 may identify managed native applications 610 that are allowed to have access to highly classified data requiring strong authentication, and ensure that access to these applications is only permitted after performing appropriate authentication, even if this means a re-authentication is required by the user after a prior weaker level of login.

[0090] Another security feature of this solution is the encryption of the data vaults 616 (containers) on the mobile device 602. The vaults 616 may be encrypted so that all on-device

data including files, databases, and configurations are protected. For on-line vaults, the keys may be stored on the server (gateway server 606), and for off-line vaults, a local copy of the keys may be protected by a user password or biometric validation. If or when data is stored locally on the mobile device 602 in the secure container 616, it may be preferred that a minimum of AES 256 encryption algorithm be utilized.

[0091] Other secure container features may also be implemented. For example, a logging feature may be included, wherein security events happening inside a managed application 610 may be logged and reported to the backend. Data wiping may be supported, such as if or when the managed application 610 detects tampering, associated encryption keys may be written over with random data, leaving no hint on the file system that user data was destroyed. Screenshot protection may be another feature, where an application may prevent any data from being stored in screenshots. For example, the key window's hidden property may be set to YES. This may cause whatever content is currently displayed on the screen to be hidden, resulting in a blank screenshot where any content would normally reside.

[0092] Local data transfer may be prevented, such as by preventing any data from being locally transferred outside the application container, e.g., by copying it or sending it to an external application. A keyboard cache feature may operate to disable the autocorrect functionality for sensitive text fields. SSL certificate validation may be operable so the application specifically validates the server SSL certificate instead of it being stored in the keychain. An encryption key generation feature may be used such that the key used to encrypt data on the mobile device 602 is generated using a passphrase or biometric data supplied by the user (if offline access is required). It may be XORed with another key randomly generated and stored on the server side if offline access is not required. Key Derivation functions may operate such that keys generated from the user password use KDFs (key derivation functions, notably Password-Based Key Derivation Function 2 (PBKDF2)) rather than creating a cryptographic hash of it. The latter makes a key susceptible to brute force or dictionary attacks.

[0093] Further, one or more initialization vectors may be used in encryption methods. An initialization vector will cause multiple copies of the same encrypted data to yield different cipher text output, preventing both replay and cryptanalytic attacks. This will also prevent an attacker from decrypting any data even with a stolen encryption key. Further, authentication then decryption may be used, wherein application data is decrypted only after the user has authenticated within the application. Another feature may relate to sensitive data in memory,

which may be kept in memory (and not in disk) only when it's needed. For example, login credentials may be wiped from memory after login, and encryption keys and other data inside objective-C instance variables are not stored, as they may be easily referenced. Instead, memory may be manually allocated for these.

[0094] An inactivity timeout may be implemented, wherein after a policy-defined period of inactivity, a user session is terminated.

[0095] Data leakage from the application management framework 614 may be prevented in other ways. For example, if or when a managed application 610 is put in the background, the memory may be cleared after a predetermined (configurable) time period. When backgrounded, a snapshot may be taken of the last displayed screen of the application to fasten the foregrounding process. The screenshot may contain confidential data and hence should be cleared.

[0096] Another security feature may relate to the use of an OTP (one time password) 620 without the use of an AD (active directory) 622 password for access to one or more applications. In some cases, some users do not know (or are not permitted to know) their AD password, so these users may authenticate using an OTP 620 such as by using a hardware OTP system like SecurID (OTPs may be provided by different vendors also, such as Entrust or Gemalto). In some cases, after a user authenticates with a user ID, a text may be sent to the user with an OTP 620. In some cases, this may be implemented only for online use, with a prompt being a single field.

[0097] An offline password may be implemented for offline authentication for those managed applications 610 for which offline use is permitted via enterprise policy. For example, an enterprise may want StoreFront to be accessed in this manner. In this case, the client agent 604 may require the user to set a custom offline password and the AD password is not used. Gateway server 606 may provide policies to control and enforce password standards with respect to the minimum length, character class composition, and age of passwords, such as described by the standard Windows Server password complexity requirements, although these requirements may be modified.

[0098] Another feature may relate to the enablement of a client side certificate for certain applications 610 as secondary credentials (for the purpose of accessing PKI protected web resources via the application management framework micro VPN feature). For example, a managed application 610 may utilize such a certificate. In this case, certificate-based

authentication using ActiveSync protocol may be supported, wherein a certificate from the client agent 604 may be retrieved by gateway server 606 and used in a keychain. Each managed application 610 may have one associated client certificate, identified by a label that is defined in gateway server 606.

[0099] Gateway server 606 may interact with an enterprise special purpose web service to support the issuance of client certificates to allow relevant managed applications to authenticate to internal PKI protected resources.

[0100] The client agent 604 and the application management framework 614 may be enhanced to support obtaining and using client certificates for authentication to internal PKI protected network resources. More than one certificate may be supported, such as to match various levels of security and/or separation requirements. The certificates may be used by the Mail and Browser managed applications 610, and ultimately by arbitrary wrapped applications 610 (provided those applications use web service style communication patterns where it is reasonable for the application management framework to mediate HTTPS requests).

[0101] Application management client certificate support on iOS may rely on importing a public-key cryptography standards (PKCS) 12 BLOB (Binary Large Object) into the iOS keychain in each managed application 610 for each period of use. Application management framework client certificate support may use a HTTPS implementation with private in-memory key storage. The client certificate may not be present in the iOS keychain and may not be persisted except potentially in “online-only” data value that is strongly protected.

[0102] Mutual SSL or TLS may also be implemented to provide additional security by requiring that a mobile device 602 is authenticated to the enterprise, and vice versa. Virtual smart cards for authentication to gateway server 606 may also be implemented.

[0103] Both limited and full Kerberos support may be additional features. The full support feature relates to an ability to do full Kerberos login to Active Directory (AD) 622, using an AD password or trusted client certificate, and obtain Kerberos service tickets to respond to HTTP Negotiate authentication challenges. The limited support feature relates to constrained delegation in Citrix Access Gateway Enterprise Edition (AGEE), where AGEE supports invoking Kerberos protocol transition so it can obtain and use Kerberos service tickets (subject to constrained delegation) in response to HTTP Negotiate authentication challenges. This mechanism works in reverse web proxy (aka corporate virtual private

network (CVPN)) mode, and when HTTP (but not HTTPS) connections are proxied in VPN and MicroVPN mode.

[0104] Another feature may relate to application container locking and wiping, which may automatically occur upon jail-break or rooting detections, and occur as a pushed command from administration console, and may include a remote wipe functionality even when a managed application 610 is not running.

[0105] A multi-site architecture or configuration of enterprise application store and an application controller may be supported that allows users to be serviced from one of several different locations in case of failure.

[0106] In some cases, managed applications 610 may be allowed to access a certificate and private key via an API (for example, OpenSSL). Trusted managed applications 610 of an enterprise may be allowed to perform specific Public Key operations with an application's client certificate and private key. Various use cases may be identified and treated accordingly, such as if or when an application behaves like a browser and no certificate access is required, if or when an application reads a certificate for "who am I," if or when an application uses the certificate to build a secure session token, and if or when an application uses private keys for digital signing of important data (e.g. transaction log) or for temporary data encryption.

[0107] OPTIMIZED NETWORK SELECTION

[0108] Computer software, hardware, and networks may be utilized in a variety of different system environments, including standalone, networked, remote-access (also known as remote desktop), virtualized, and/or cloud-based environments, among others. For example, the network architectures of FIG. 5 and/or FIG. 6 may be used to implement one or more illustrative aspects described herein.

[0109] FIG. 7 illustrates one example of a general workflow of network access between an electronic device 700 and an application server 702 that may be used to implement one or more illustrative aspects described herein in a standalone and/or networked environment. An electronic device 700 (interchangeably referred to as "communication device 700") may be a wireless communication device, such as a laptop computer, tablet computer, smart phone, a notebook computer, a netbook computer, or other type of communication device, e.g., device 502 and/or 602 as alternatively shown in FIG. 5 and FIG. 6, respectively. The communication device 700 may include, but is not restricted to, a processor for controlling overall operations

of the communication device 700, a random access memory (RAM), read only memory (ROM), network interface, input/output interfaces (e.g., keyboard, mouse, display, printer, etc.), and memory. The input/output (I/O) may include a variety of interface units and drives for reading, writing, displaying, and/or printing data or files. The memory may further store operating system software, control logic, and other application software providing secondary, support, and/or other functionality which may or might not be used in conjunction with aspects described herein.

[0110] The application server 702 may be any type of known computer, server, or data processing device. In one embodiment, application server 702 may be configured as any type of server, as needed, e.g., a file server, an application server, a web server, a proxy server, an appliance, a network appliance, a gateway, an application gateway, a gateway server, a virtualization server, a deployment server, a Secure Sockets Layer (SSL) VPN server, a firewall, a web server, an application server or as a master application server, a server executing an active directory, or a server executing an application acceleration program that provides firewall functionality, application functionality, or load balancing functionality. Other server types may also be used.

[0111] Application server 702 e.g., may include a processor for controlling overall operation of the application server 702. Application server 702 may further include random access memory (RAM), read only memory (ROM), network interface, input/output interfaces (e.g., keyboard, mouse, display, printer, etc.), and memory. Input/output (I/O) may include a variety of interface units and drives for reading, writing, displaying, and/or printing data or files. Memory may further store operating system software for controlling overall operation of the application server 702, control logic for instructing application server 702 to perform aspects described herein, and other application software providing secondary, support, and/or other functionality which may or might not be used in conjunction with aspects described herein.

[0112] In some embodiments, application server 702 may be a web server, a proxy server, an appliance, a network appliance, a gateway, an application gateway, and so forth. The application server 702 may be configured to provide data related to a software application. For example, a social media application requires access to a social media server to fetch social media data.

[0113] The communication device 700 may further include a managed application 701 for managing a communication network between the communication device 700 and the application server 702. The managed application 701 may aid in determining and selecting an optimized communication network between the communication device 700 and the application server 702. Further, the managed application 701 may build a secure data transfer channel between the communication device 700 and the application server 702 to communicate and/or share the data. In one embodiment, the managed application 701 may select an optimized communication network from a list of available communication networks based on one or more policies over which the communication device 700 may communicate with the application server 702. Each policy may be defined in accordance with an enterprise mobility management system, e.g., enterprise mobility technical architecture 500 and the like, such that an enterprise or other controlling entity can determine what data and under what circumstances a user device can and cannot use various networks. For example, a user device might only be permitted to access enterprises resources over a wireless network that meets minimum security thresholds. Alternatively, a user might only be able to access a corporate wireless network from one or more geofenced areas or times of day.

[0114] In another embodiment, the managed application 701 may dynamically select an optimized communication network over which the communication device 700 may communicate with the application server 702 based on network probing. For example, device 700 may dynamically assess one or more network characteristics of each available wireless network, and determine which to use based a set of heuristics and the network characteristics. For example, the heuristics might indicate preferred networks based on data security, network speed, data cost, signal strength, etc.

[0115] In yet another embodiment, the managed application 701 may select an optimized communication network based both on policies as well as using dynamic probing. The managed application 701 may be a software application stored in the memory of the communication device 700, and may first attempt to use policy based network selection which, if unsuccessful, then performs dynamic selection based on the network probe.

[0116] Further, the communication device 700 may also include software applications such as, but is not restricted to, an ecommerce application, a package tracking/reading application, a location-based service application, a navigation application, a content provisioning application, a camera/imaging application, a media player application, a social networking application, and the like.

[0117] In one embodiment, the communication network may be a wireless communication network. The communication network may be a wide area network (WAN), such as the Internet. Other networks may also or alternatively be used, including private intranets, corporate networks, local area networks (LAN), metropolitan area networks (MAN), wireless networks, personal networks (PAN), and the like. Communication network may be replaced with fewer or additional computer networks. A local area network may have one or more of any known LAN topology and may use one or more of a variety of different protocols, such as Ethernet. The communication device 700 and the application server 702 and other devices (not shown) may be connected to one or more of the communication networks via twisted pair wires, coaxial cable, fiber optics, radio waves, or other communication media. In some embodiments, the wireless network may employ various technologies including, for example, Code Division Multiple Access (CDMA), Enhanced Data Rates For Global Evolution (EDGE), General Packet Radio Service (GPRS), Mobile Ad Hoc Network (MANET), Global System For Mobile Communications (GSM), 3G, 4G, 5G, Long-Term Evolution (LTE), Internet Protocol Multimedia Subsystem (IMS), Universal Mobile Telecommunications System (UMTS), etc., as well as any other suitable wireless medium, e.g., microwave access (WiMAX), Wireless Fidelity (Wi-Fi), satellites, Wireless LAN (WLAN), Bluetooth®, Internet Protocol (IP) data casting, and so forth.

[0118] The term “network” as used herein and depicted in the drawings refers not only to systems in which remote storage devices are coupled together via one or more communication paths, but also to stand-alone devices that may be coupled, from time to time, to such systems that have storage capability. Consequently, the term “network” includes not only a “physical network” but also a “content network,” which is comprised of the data attributable to a single entity which resides across all physical networks.

[0119] In one embodiment, a user of the communication device 700 may interact with the application server 702 by using a software application. In some embodiments, a user of the communication device 700 may interact with the application server 702 by using a web browser to share data with the application server 702 via one or more externally exposed websites hosted by a web server. For example, from a communication device 700, a user may access the application server 702 by using an Internet browser, as is known in the art, or by executing a software application that communicates with the application server 702 over a communication network.

[0120] When a user accesses a software application to communicate with the application server 702, then the managed application 701 may initiate managing a communication network by sending and/or receiving data of the software application to and/or from the application server 702 over an optimized communication network. The managed application 701 may manage the communication network by selecting an optimized communication network for sharing the data between the communication device 700 and the application server 702.

[0121] Further, managed application 701 may communicate with the application server 702 through, but not restricted to, a Wi-Fi network 703, a cellular network provided by a first Internet Service Provider (ISP1) 704, or by a cellular network provided by a second Internet Service Provider (ISP2) 705. The selection of an optimized communication network by the managed application 701 through which the communication device 700 may communicate with the application server 702 may be based on policy control mechanism. In some embodiments, the selection of an optimized communication network by the managed application 701 through which the communication device 700 may communicate with the application server 702 may be based on dynamic control mechanism. A more detailed explanation of the architecture of selecting an optimized communication network is explained in conjunction with FIG. 10 and FIG. 11, below.

[0122] FIG. 8 illustrates one example of an architecture of selecting a communication network by the managed application 701 that may be used to implement one or more illustrative aspects described herein in a standalone and/or networked environment. The managed application 701 of the communication device 700 may communicate with the application server 702 over an optimized communication network that is selected based on, but not restricted to, a policy control mechanism 801 or a dynamic control mechanism 802.

[0123] The policy control mechanism 801 may be used by the managed application 701 to select an optimized communication network between the communication device 700 and the application server 702 based on one or more policies. A policy may be defined as a plan of action to be taken when certain conditions are met for selecting an optimized communication network for a software application to communicate with a server associated with the software application. In one embodiment, the condition associated with the communication device 700 may be determined based on contextual evaluation. In some embodiments, a policy may be defined for each software application executing on the

communication device 700. In one illustrative embodiment, the policy for each software application may be defined by a user of the communication device 700.

[0124] FIG. 9 illustrates one example of a user interface of the communication device 700 in which a user may select an optimized communication network that may be used to implement one or more illustrative aspects described herein in a standalone and/or networked environment. The user of the communication device 700 may define preferences of executing a software application over a communication network. As shown in FIG. 9, the user may select that a web browser may communicate over a cellular network provided by an Internet Service Provider 1 rather than Wi-Fi. In another illustrative scenario, the user may select that the phone contacts may only communicate over a cellular network provided by an Internet Service Provider 2 rather than the Internet Service Provider 1 or Wi-Fi. Only two ISP selection check boxes being shown in FIG. 9 is for illustration purpose only. However, more than two ISP selection check boxes may be used in case the communication device 700 has more than two physical SIM card slots. In some illustrative embodiments, the communication device 700 may have multiple physical slots for accommodating multiple SIM cards. In some other embodiments, the communication device 700 may use an embedded SIM (eSIM) card.

[0125] Each policy may be based on a number of variables such as, but not restricted to, device owner, device location, date, time, application name, application catalog, application reputation, connected Wi-Fi information (such as, Wi-Fi name, Wi-Fi type, Wi-Fi reputation, Wi-Fi encryption level, Wi-Fi signal strength, Wi-Fi authentication method, etc.), user login, user privileges, and so forth. In an illustrative scenario, a condition associated with the communication device 700 may be defined by using the variables such as an official email, corporate Wi-Fi, smartphone, such that an official email can only be sent over a corporate Wi-Fi from a smartphone. Based on the condition, a policy is defined such that when the smartphone is connected to the corporate Wi-Fi, then the official email can be sent. In case, the smartphone is connected to a cellular network, then as per the defined policy, the managed application 101 may switch from the corporate Wi-Fi to the cellular network and may then send the official email to a recipient.

[0126] As shown in Table 1, below, a software application such as a social media application (e.g., WeChat™ or Facebook™) having a social application catalog, may have a condition that the communication device 700 on which the social media application is running is currently connected to a corporate Wi-Fi during business hours. When a user of the communication device 700 desires to communicate with a social media application server

then a policy action is defined that the social media application can only transfer data with the application server 702 on a cellular network provided by an ISP 1. In this case, the managed application 701 may then switch from the communication network Wi-Fi to the cellular network ISP 1 and then transmit the data over the cellular network ISP 1. Similarly, a secure email having a productivity application catalog may have a condition that the communication device 700 is currently connected to a corporate Wi-Fi during business hours. So, when the user desires to send an official email, then a policy may be defined that the official email may be sent over the corporate Wi-Fi only. The managed application 101 may then send the email over the corporate Wi-Fi. In the same way, a banking application having a financial application catalog may have a condition that the user prefers to use cellular network for conducting financial transactions. Then a policy may be defined that when the user is executing a financial software application, then the data may be shared to a financial application server over a cellular network only either provided by ISP1 or ISP2. Therefore, the managed application 701 may select an optimized communication network as per the policy control mechanism defined by the user of the communication device 700 to communicate with the application server 702.

Table 1

Managed Application	Application Catalog	Context Evaluation (Condition)	Policy Action
WeChat or Facebook	Social	Connected to Corporate Wi-Fi during business hour	Cellular ISP 1
Secure Mail	Productivity	Connected Wi-Fi reputation is low due to reported privacy tracking	Cellular ISP 2
Secure Mail	Productivity	Connected to Corporate Wi-Fi during business hour	Use enterprise Wi-Fi
Secure Web	Productivity	Connected Wi-Fi reputation is no data but Wi-Fi authentication and encryption is weak	Cellular ISP 2
Workspace APP	Productivity	Wi-Fi info such SSID belongs to commercial provider	Cellular ISP 1
MyBankAPP	Finance	User preference: cellular	Cellular ISP 1 or ISP 2

[0127] Table 1 shows an illustrative policy control for communication network(s) associated with each software application.

[0128] The dynamic control mechanism 802 may be used by the managed application 101 to dynamically select a communication network between the communication device 700 and the application server 702. The dynamic control mechanism 802 (interchangeably

referred to as “dynamic probing”) may determine information associated with each available wireless communication network. The information associated with each available wireless communication network may include, but not restricted to, delay, cost of traffic, security such as data in transit (3G link, 4G link, 5G link, Wi-Fi, etc.), Wi-Fi information, and so forth. The Wi-Fi information may include, but not restricted to, Wi-Fi name, Wi-Fi type, Wi-Fi reputation, Wi-Fi encryption level, Wi-Fi signal strength, Wi-Fi authentication method, and other Wi-Fi security settings. In some embodiments, the information associated with each available wireless communication network may be updated periodically, such as every minute, every 5 minutes, every 10 minutes, every 1 hour, and so forth.

[0129] In Table 2, below, a software application such as a social media application (e.g., WeChat™ or Facebook™) having a condition that the communication device 700 on which the social media application is running has no access to the application server on Wi-Fi. The managed application 701 may then push a probed uniform resource locator (URL) to the application server 702 such as a social media application server. The probe URL may be pushed to the social media application to detect one or more conditions of the communication device 700. Based on the detection, the managed application 701 may select an optimized communication network to share social media data between the social media application and the social media application server. For example, when a user of the communication device 700 desires to communicate with a social media application server, the managed application 701 sends a probe URL (e.g., Test.Facebook.com) to the social media application server to test the condition of the communication device 700. The test result may show that the social media application cannot be accessed on Wi-Fi, then the managed application 701 may select a cellular data to share social media data with the social media application server provided by an ISP 1. Likewise, when the user desires to send an email, then a probe URL (e.g., mail.citrix.com) is pushed to a web server. It may be determined that the latency and round trip delay (RTT) to the probe URL is higher from Wi-Fi or cellular network provided by ISP 1 than cellular network provided by ISP 2. Based on this condition, a policy action may be defined that the email may be sent over the cellular network provided by ISP 2. Then, the managed application 701 may select the cellular network provided by ISP 2 to send the email to a recipient.

[0130] As another example, when the user desires to access a website, it may be determined that the connected Wi-Fi authentication and encryption is weak and the cellular network provided by ISP 2 is 4G while cellular network provided by ISP 1 is 3G. Based on

the conditions, a policy action may be defined that the website may be accessed over the cellular network provided by the ISP 2. The managed application 101 may then share data of the website over the cellular network provided by the ISP 2. In another illustrative scenario, a user of the communication device 700 may access a workspace application having a productivity application catalog. The managed application 701 may then determine that the Wi-Fi connected to the communication device 700 belongs to a commercial provider based on a Service Set Identifier (SSID) associated with the communication network. Then, the managed application 701 may establish a communication channel between the communication device 700 and the application server 702 on a cellular network provided by the service provider ISP 1.

[0131] The managed application 701 is, therefore, configured to establish a data transfer communication channel between the communication device 700 and the application server 702 to transfer data.

Table 2

Managed Application	Pushing Probed URL to app	Context Evaluation (Condition)	Policy Action
WeChat or Facebook	Test.Facebook.com	Wi-Fi no access to Facebook	Cellular ISP1
Secure Mail	Mail.citrix.com	Latency and RTT to probe URL is higher from Wi-Fi or cellular ISP1 than cellular ISP2	Cellular ISP2
Secure Web	m.mycompany.com	Connected Wi-Fi reputation is no data but Wi-Fi authentication and encryption is weak , AND, Cellular ISP2 is 4G than ISP1 is 3G (4G using higher encryption than 3G)	Cellular ISP2

[0132] Table 2 shows an illustrative dynamic control for communication network associated with each software application.

[0133] FIG. 10 illustrates one example of an architecture of network access policy control that may be used to implement one or more illustrative aspects described herein in a standalone and/or networked environment. The architecture may be shown to select an optimized communication network between the communication device 700 and the application server 702 based on policies. The architecture includes a communication device

1000 (e.g., device 700) that may share data with an application server 7002 (e.g., server 702) over a communication network. As discussed above, the communication device 1000 may illustratively be a wireless communication device. The application server 1002 may be configured to provide data related to a software application to the communication device 1000.

[0134] The communication device 1000 may include a managed application 1001 configured to select an optimized communication network to share data between the communication device 1000 and the application server 1002. In addition, the communication device 1000 may include other software applications such as, but is not restricted to, an ecommerce application, a package tracking/reading application, a location-based service application, a navigation application, a content provisioning application, a camera/imaging application, a media player application, a social networking application, and the like.

[0135] The communication device 1000 may further include a policy engine 1003 for managing a communication network between the communication device 1000 and application server 1002. In one embodiment, the policy engine 1003 may be built in the managed application 1001. The policy engine 1003 may be configured to select an optimized communication network from a list of available communication networks based on policies. As discussed above, a policy may be defined for each software application executing on the communication device 1000. In one embodiment, the policy for each software application may be defined by a user of the communication device 1000.

[0136] The policy engine 1003 may receive a request 1005 from the managed application 1001 for communication network access to share data with the application server 1002. In one embodiment, the policy engine 1003 may receive the request 1005 when a user clicks on a software application to execute it on the communication device 1000. In other embodiments, the request 1005 may be sent when the user accesses a website on a web browser in the communication device 1000. Based on the received request, the policy engine 1003 may retrieve condition relation information 1004 associated with the communication device 1000. In some embodiments, the condition related information may be retrieved from a memory such as a memory of the communication device 1000. In other embodiments, the condition related information may be retrieved from one or more remote databases and/or data stores. The condition related information associated with the communication device 1000 may include, but not restricted to, a device owner, device location, date, time, application name, application catalog, application reputation, encryption and/or

authentication method, security type, connected Wi-Fi information (such as, Wi-Fi name, Wi-Fi type, Wi-Fi reputation, Wi-Fi encryption level, Wi-Fi signal strength, Wi-Fi authentication method, etc.), user login, user privileges, user preference, and so forth.

[0137] The policy engine 1003 may process the condition related information 1004 to determine and select an optimized communication network (e.g., cellular or Wi-Fi) for the communication device 1000. In one embodiment, the policy engine 1003 may process the condition related information 1004 based on a context based evaluation. Based on the processing, the policy engine 1003 may select an optimized communication network from a list of available communication networks.

[0138] Further, the policy engine 1003 may request 1006 the application server 1002 to transfer and/or share data over the selected communication network. In one embodiment, the managed application 1001 of the communication device 1000 may then transmit a request 1006 to the application server 1002 over the selected communication network to transmit the data over the same selected communication network. In response to the request 1006, the application server 1002 may then send the requested data 1007 to the communication device 1000 and the managed application 1001 over the selected communication network. Therefore, a communication channel between the communication device 1000 and the application server 1002 is established to transmit data.

[0139] FIG. 11 illustrates one example of an architecture of network access dynamic control that may be used to implement one or more illustrative aspects described herein in a standalone and/or networked environment. The architecture includes a communication device 1100 that may request data from a computing device data center 1102 over a communication network. In one embodiment, the computing device data center 1102 may be a policy server such as the policy engine 1003. In one embodiment, the communication device 1100 may be a wireless communication device, such as, a laptop, a notebook computer, a tablet computer, and so forth. The computing device data center 1102 may be configured to provide data related to a managed application 1101. The communication device 1100 and the computing device data center 1102 may be connected through a physical connection or through a wireless connection.

[0140] The communication device 1100 may include the managed application 1101 that may need to request data from the computing device data center 1102. The managed application 1101 may aid in selecting an optimized communication network between a

software application of the communication device 1100 and a data center. The communication device 1100 may also include, software applications such as, but is not restricted to, an ecommerce application, a package tracking/reading application, a location-based service application, a navigation application, a content provisioning application, a camera/imaging application, a media player application, a social networking application, and the like.

[0141] The managed application 1101 of the communication device 1100 may further include a network access probe 1103. In one embodiment, the network access probe 1103 may be built in the managed application 1101. The network access probe 1103 may be configured to dynamically select an optimized communication network for sharing data between the managed application 101 of the communication device 1100 and the computing device data center 1102. The network access probe 1103 may receive probe related information 1108 when a software application is initiated in the communication device 1100. In one embodiment, the probe related information may be a signal received by the network access probe 1103 when a software application such as a banking application is initiated by the user of the communication device 1100. When the probe related information 1108 is received by the network access probe 1103, then the network access probe 1103 may test quality of each available communication network between the communication device 1100 and the computing device data center 1102. In order to test the quality of each communication network, the network access probe 1103 may send a probe 1104 to the computing device data center 1102. In one embodiment, the network access probe 1103 may send a probe such as a test URL to the computing device data center 1102 over an available communication network. In some embodiments, the network access probe 1103 may send an empty message to the computing device data center 1102 over an available communication network.

[0142] When the probe is received by the computing device data center 1102, a probe result 1105 is sent back by the computing device data center 1102 to the network access probe 1103. In one embodiment, the probe result 1105 of the available communication network may include test information such as, but not restricted to, network delay, cost, safety, Wi-Fi reputation, signal strength, and so forth, associated with the communication network. In some embodiments, the probe results 1105 may also include policies related to the communication device 1100 and/or software application. Further, based on the network test result, the managed application 1101 may select an optimized communication network

for communication between the communication device 1100 and the computing device data center 1102. The managed application 1101 may then request 1106 the computing device data center 1102 to transfer data over the optimized communication network. The computing device data center 1102 may then transfer the requested data 1107 over the optimized communication network to the managed application 1101 by building a transfer channel between the communication device 1100 and the computing device data center 1102.

[0143] FIG. 12 depicts a flowchart that illustrates a method of selecting an optimized communication channel to transmit data between a communication device and an application server in accordance with one or more illustrative aspects discussed herein. In one or more embodiments, the method illustrates in FIG. 12 and/or one or more steps thereof may be performed by a communication device. In other embodiments, the method illustrated in FIG. 12 and/or one or more steps thereof may be embodied in computer-executable instructions that are stored in a computer-readable medium, such as a non-transitory computer-readable memory.

[0144] As seen in FIG. 12, in step 1801, a managed application in the communication device (e.g., a wireless communication device, such as a laptop computer, tablet computer, smart phone, or other type of communication device) may determine whether a software application running on the communication device is trying to access a wireless communication channel or network. The software application, may be, for instance, an email application, a web browsing application, social media application, financial application, and so forth.

[0145] In one or more embodiments, the managed application may receive a request from a software application executing on the communication device to transmit data to a first recipient. In addition, the managed application may receive another request from another software application executing on the communication device to transmit data to a second recipient. In an illustrative scenario, a user of the communication device desires to access a social media application, and the user also desires to send a business email to a client.

[0146] Next, in step 1802, the managed application may determine whether the software application is trying to access a communication network. The communication network may be a wireless communication network. Various communication networks may include a wide area network (WAN), such as the Internet, private intranets, corporate networks, local area networks (LAN), metropolitan area networks (MAN), wireless networks, personal networks

(PAN), cellular network, and the like. The communication device may be connected to one or more of the communication networks via twisted pair wires, coaxial cable, fiber optics, radio waves, or other communication media. In one or more embodiments, the wireless network may be conforming to an IEEE 802.11. standard.

[0147] The software application executing on the communication device may be trying to access the communication network for receiving and/or transmitting data to another device. For example, in the above exemplary scenario, the social media application is trying to fetch social media data from remote social media data centers/servers.

[0148] Further, in step 1802, if the managed application determines that the software application of the communication device is not trying to access the communication network, then the method returns to step 1801 and continues to monitor whether a software application is trying to access the communication network. In case, if the managed application determines that the software application is trying to access the communication network, then the method proceeds towards step 1203.

[0149] In step 1203, the managed application determines given conditions associated with the communication device. In one embodiment, the condition associated with the communication device 700 may be determined based on contextual evaluation. In some embodiments, the conditions may be predefined by an application developer of the software application. The given conditions may include variable, for example, device owner, device location, date, time, application name, application catalog, application reputation, connected Wi-Fi information (such as, Wi-Fi name, Wi-Fi type, Wi-Fi reputation, Wi-Fi encryption level, and Wi-Fi authentication method, etc.), user login, user privileges, and so forth. Based on the conditions, a policy may be defined. In the above exemplary scenario, the communication device trying to fetch social media data is currently connected to a corporate Wi-Fi, then the managed application may define a policy that the social media application can only transmit data on a cellular network, while a secure official email can be sent through the corporate Wi-Fi connected to the communication device.

[0150] In step 1204, the managed application determines whether the given condition exists. When the managed application determines that the given condition associated with the communication device does not exist, then the method proceeds to a step 1205. When the managed application determines that the given condition associated with the communication device do exist, then the method proceeds to a step 1206.

[0151] In step 1205, the managed application dynamically controls the available communication network via a probe, e.g., as discussed above. The managed application may send a probe to test the quality and other factors of the communication networks between the communication device and the application server. The test may show results such as, but are not restricted to, network delay, cost, safety, Wi-Fi- reputation, and so forth. Based on the test results, the managed application may select an optimized communication network (e.g., cellular or Wi-Fi) for the communication device based on the test results to transmit and/or receive data with the application server.

[0152] In step 1206, the managed application selectively chooses a corresponding communication network as per the condition and transmits and/or receives the data to and/or from the application server. In the above illustrative scenario, the communication device trying to fetch social media data selects a cellular network as per the given condition and then transmits the social media data to the remote social media data centers/servers.

[0153] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are described as example implementations of the following claims.

CLAIMS

What is claimed is:

1. A method comprising:

receiving, within a wireless communications device from a first application executing on the wireless communications device, a request to transmit first data to a first recipient;

receiving, within the wireless communications device from a second application executing on the wireless communications device, a request to transmit second data to a second recipient;

determining, by the wireless communications device, that a plurality of wireless communications channels are available through which the wireless communications device can transmit the first data;

selecting, when a state of the wireless communications device satisfies an existing policy established by a policy engine executing on the wireless communications device, a wireless communication channel identified by the satisfied policy;

selecting, when the state of the wireless communications device does not satisfy any existing policy established by the policy engine, the wireless communication channel based on a dynamic probe of the plurality of available wireless communications channels;

sending the first data to the first recipient over a first wireless communications channel selected from the plurality of available wireless communications channels based on a first state of the wireless communications device satisfying an existing policy, said first state associated with the first application; and

sending the second data to the second recipient over a second wireless communications channel selected from the plurality of available wireless communications channels based on the dynamic probe of the plurality of available wireless communications channels resulting from a second state of the wireless communications device not satisfying any existing policies.

2. The method of claim 1, wherein the policy engine stores a plurality of policies, each identifying a set of one or more wireless communications channels permitted when that policy is satisfied.

3. The method of claim 1, wherein one of the plurality of wireless communications channels comprises cellular communications.

4. The method of claim 1, wherein one of the plurality of wireless communications channels comprises a wireless network conforming to an IEEE 802.11 standard.

5. The method of claim 1, wherein each policy is based on a plurality of variables comprising device owner, device location, date, time, application name, application catalog, application reputation, connected Wi-Fi info, user login, and user privileges.

6. The method of claim 5, wherein connected Wi-Fi info further comprises Wi-Fi name, Wi-Fi type, Wi-Fi reputation, Wi-Fi encryption level, and Wi-Fi authentication method.

7. The method of claim 1, wherein the dynamic probe comprises determining, for each of the plurality of available wireless communications channels, delay, cost of traffic, and security, and further comprises, for each available Wi-Fi communications channel, Wi-Fi name, Wi-Fi type, Wi-Fi reputation, Wi-Fi encryption level, and Wi-Fi authentication method.

8. One or more computer readable media storing computer readable instructions which, when executed by a processor of a wireless communication device, cause the device to perform:

receiving, from a first application executing on the wireless communications device, a request to transmit first data to a first recipient;

receiving, from a second application executing on the wireless communications device, a request to transmit second data to a second recipient;

determining that a plurality of wireless communications channels are available through which the wireless communications device can transmit the first data;

selecting, when a state of the wireless communications device satisfies an existing policy established by a policy engine executing on the wireless communications device, a wireless communication channel identified by the satisfied policy;

selecting, when the state of the wireless communications device does not satisfy any existing policy established by the policy engine, the wireless communication channel based on a dynamic probe of the plurality of available wireless communications channels;

sending the first data to the first recipient over a first wireless communications channel selected from the plurality of available wireless communications channels based on a

first state of the wireless communications device satisfying an existing policy, said first state associated with the first application; and

 sending the second data to the second recipient over a second wireless communications channel selected from the plurality of available wireless communications channels based on the dynamic probe of the plurality of available wireless communications channels resulting from a second state of the wireless communications device not satisfying any existing policies.

9. The computer readable media of claim 8, wherein the policy engine stores a plurality of policies, each identifying a set of one or more wireless communications channels permitted when that policy is satisfied.

10. The computer readable media of claim 8, wherein one of the plurality of wireless communications channels comprises cellular communications.

11. The computer readable media of claim 8, wherein one of the plurality of wireless communications channels comprises a wireless network conforming to an IEEE 802.11 standard.

12. The computer readable media of claim 8, wherein each policy is based on a plurality of variables comprising device owner, device location, date, time, application name, application catalog, application reputation, connected Wi-Fi info, user login, and user privileges.

13. The computer readable media of claim 12, wherein connected Wi-Fi info further comprises Wi-Fi name, Wi-Fi type, Wi-Fi reputation, Wi-Fi encryption level, and Wi-Fi authentication method.

14. The computer readable media of claim 8, wherein the dynamic probe comprises determining, for each of the plurality of available wireless communications channels, delay, cost of traffic, and security, and further comprises, for each available Wi-Fi communications channel, Wi-Fi name, Wi-Fi type, Wi-Fi reputation, Wi-Fi encryption level, and Wi-Fi authentication method.

15. A wireless communications device, comprising:
a processor; and
memory storing computer readable instructions which, when executed by the processor, cause the device to perform:
receiving, from a first application executing on the wireless communications device, a request to transmit first data to a first recipient;
receiving, from a second application executing on the wireless communications device, a request to transmit second data to a second recipient;
determining that a plurality of wireless communications channels are available through which the wireless communications device can transmit the first data;
selecting, when a state of the wireless communications device satisfies an existing policy established by a policy engine executing on the wireless communications device, a wireless communication channel identified by the satisfied policy;
selecting, when the state of the wireless communications device does not satisfy any existing policy established by the policy engine, the wireless communication channel based on a dynamic probe of the plurality of available wireless communications channels;
sending the first data to the first recipient over a first wireless communications channel selected from the plurality of available wireless communications channels based on a first state of the wireless communications device satisfying an existing policy, said first state associated with the first application; and
sending the second data to the second recipient over a second wireless communications channel selected from the plurality of available wireless communications channels based on the dynamic probe of the plurality of available wireless communications channels resulting from a second state of the wireless communications device not satisfying any existing policies.
16. The wireless communications device of claim 15, wherein the policy engine stores a plurality of policies, each identifying a set of one or more wireless communications channels permitted when that policy is satisfied.
17. The wireless communications device of claim 15, wherein one of the plurality of wireless communications channels comprises cellular communications, and a second of the

plurality of wireless communications channels comprises a wireless network conforming to an IEEE 802.11 standard.

18. The wireless communications device of claim 15, wherein each policy is based on a plurality of variables comprising device owner, device location, date, time, application name, application catalog, application reputation, connected Wi-Fi info, user login, and user privileges.

19. The wireless communications device of claim 18, wherein connected Wi-Fi info further comprises Wi-Fi name, Wi-Fi type, Wi-Fi reputation, Wi-Fi encryption level, and Wi-Fi authentication method.

20. The wireless communications device of claim 15, wherein the dynamic probe comprises determining, for each of the plurality of available wireless communications channels, delay, cost of traffic, and security, and further comprises, for each available Wi-Fi communications channel, Wi-Fi name, Wi-Fi type, Wi-Fi reputation, Wi-Fi encryption level, and Wi-Fi authentication method.

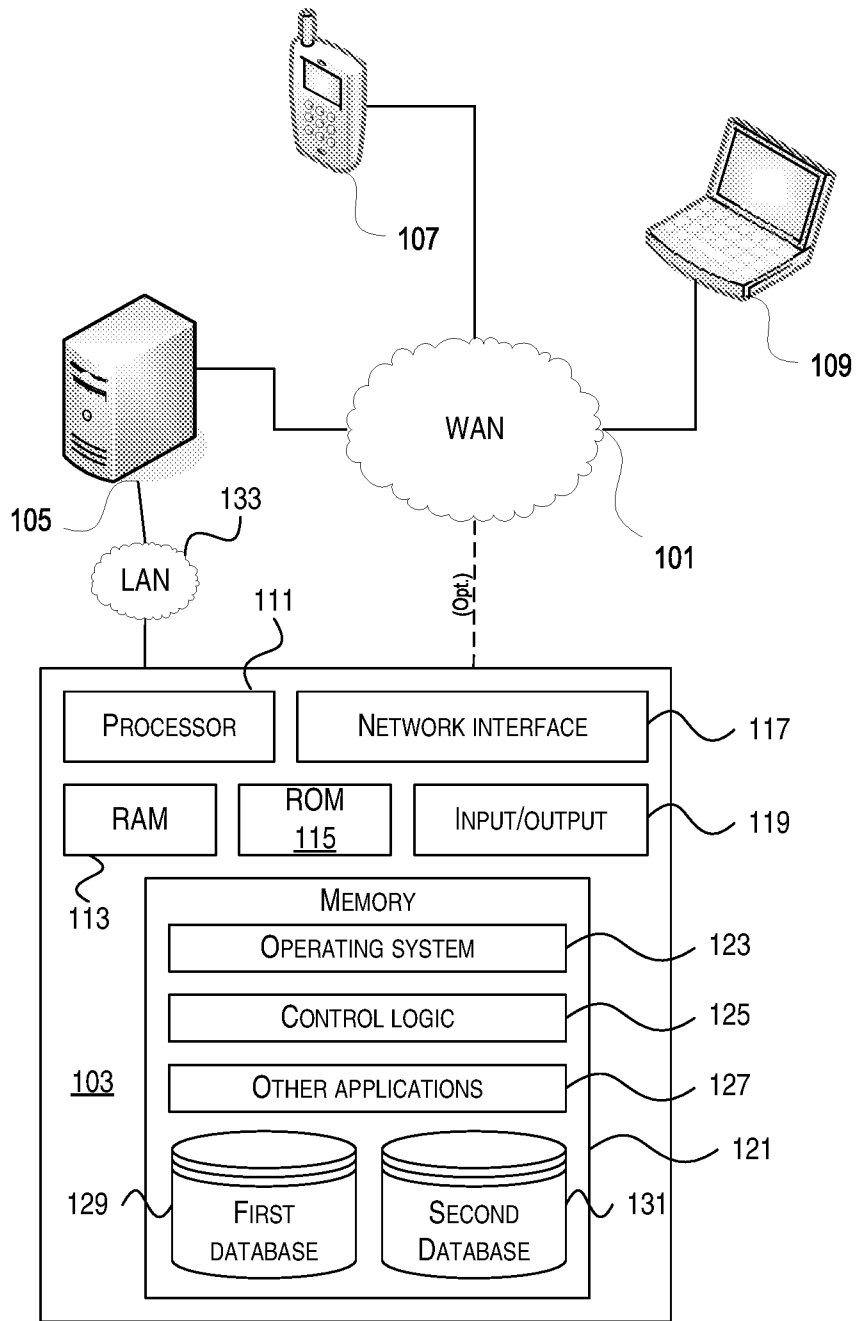


FIG. 1



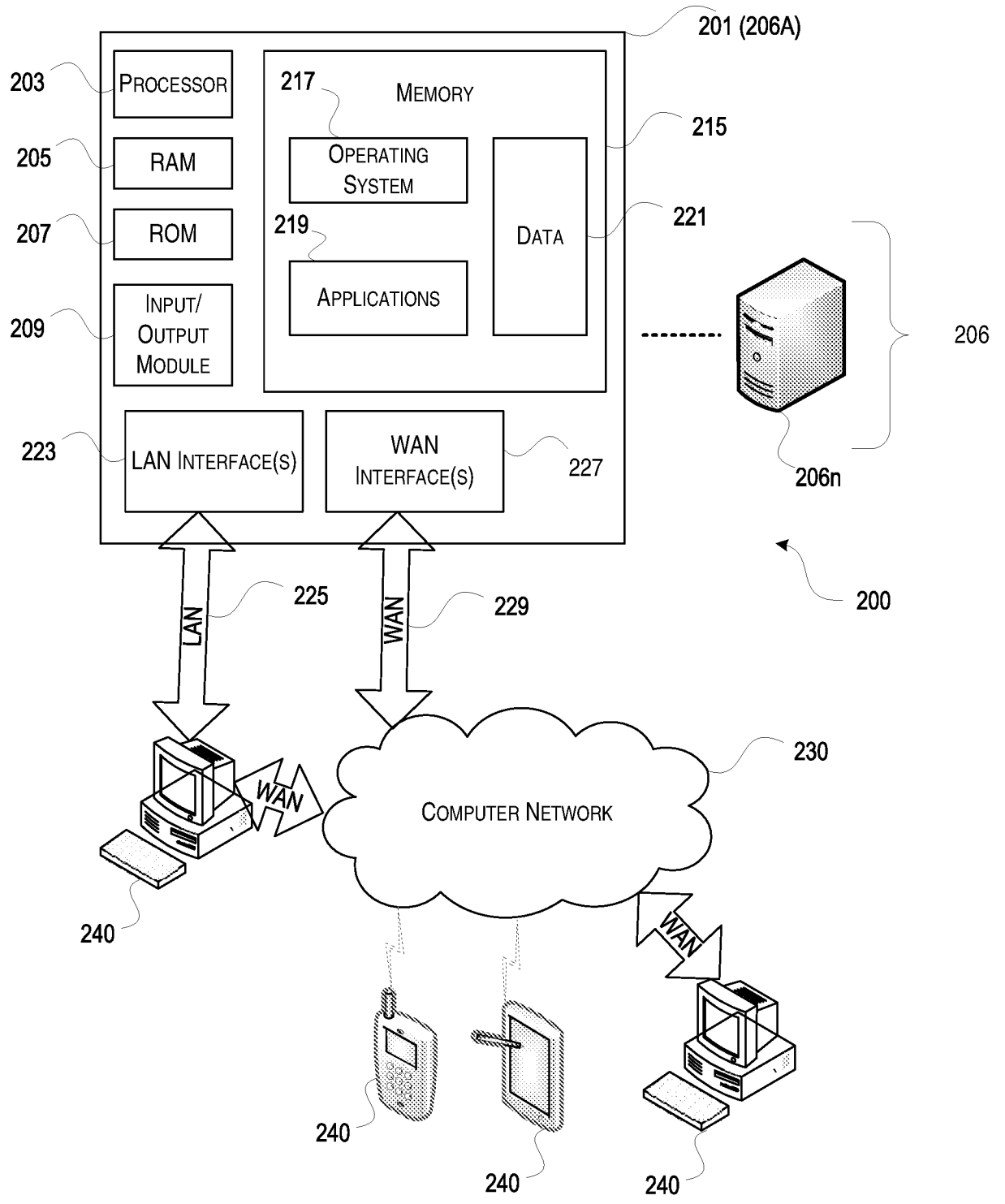


FIG. 2



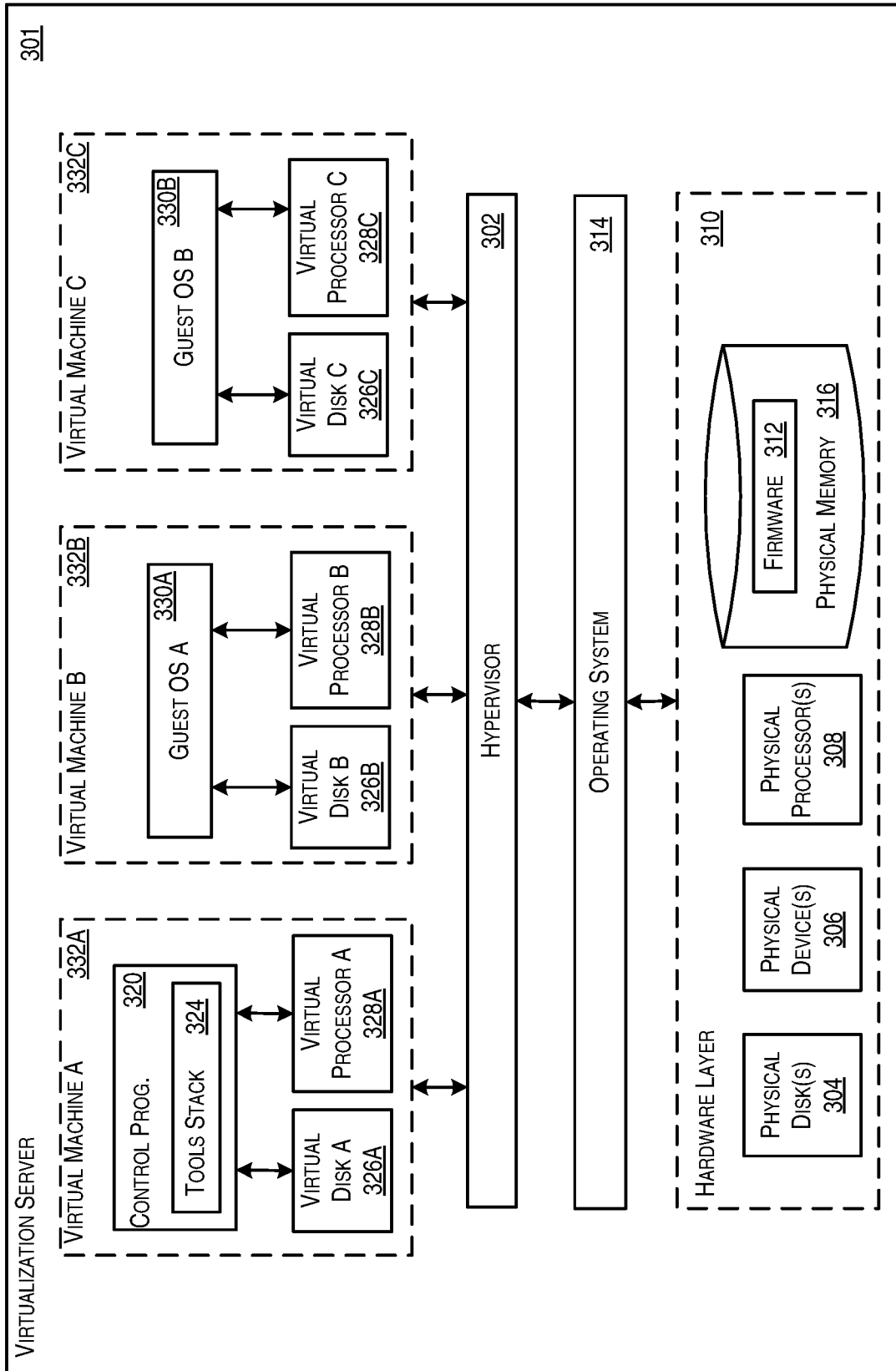


FIG. 3



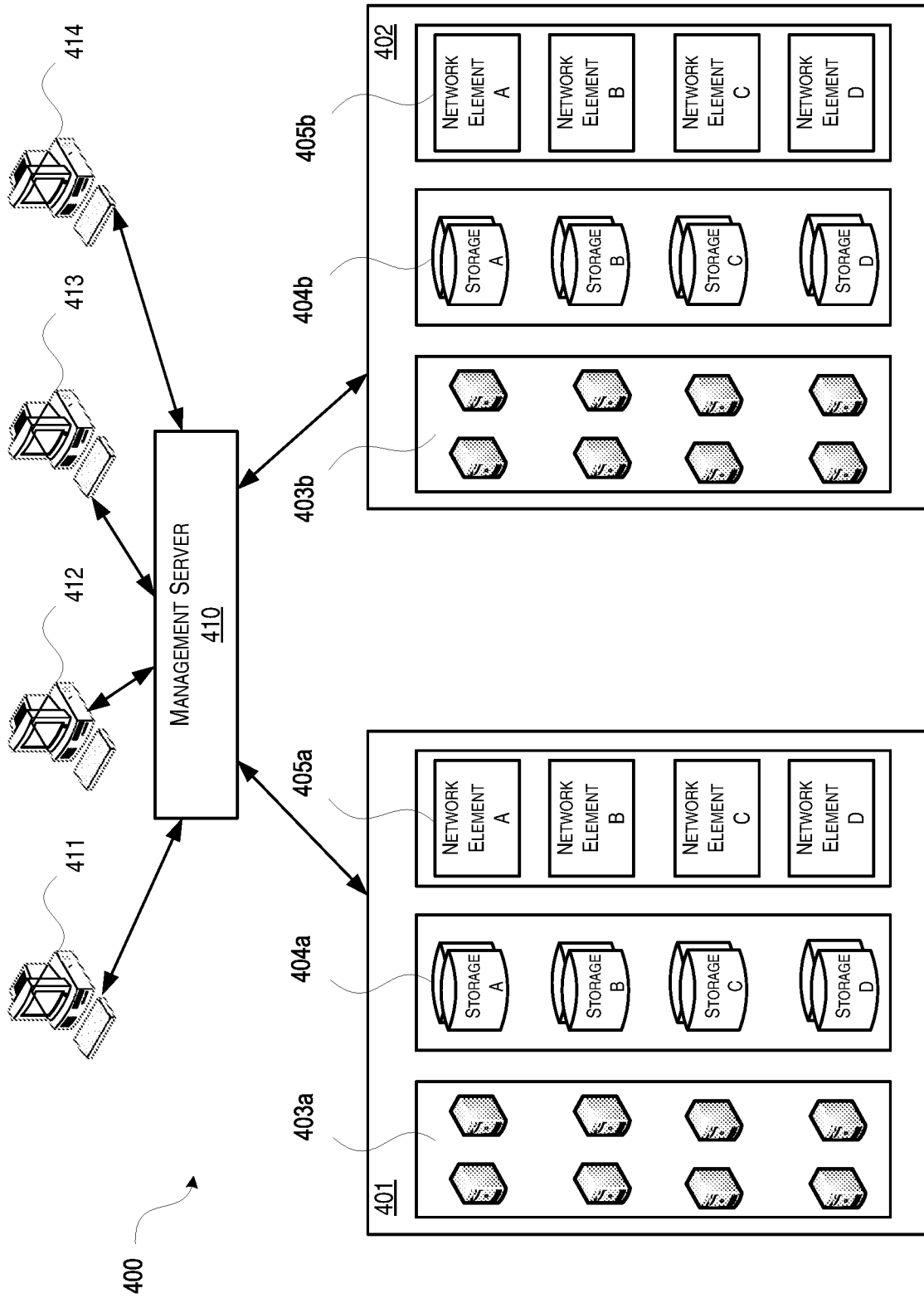


FIG. 4



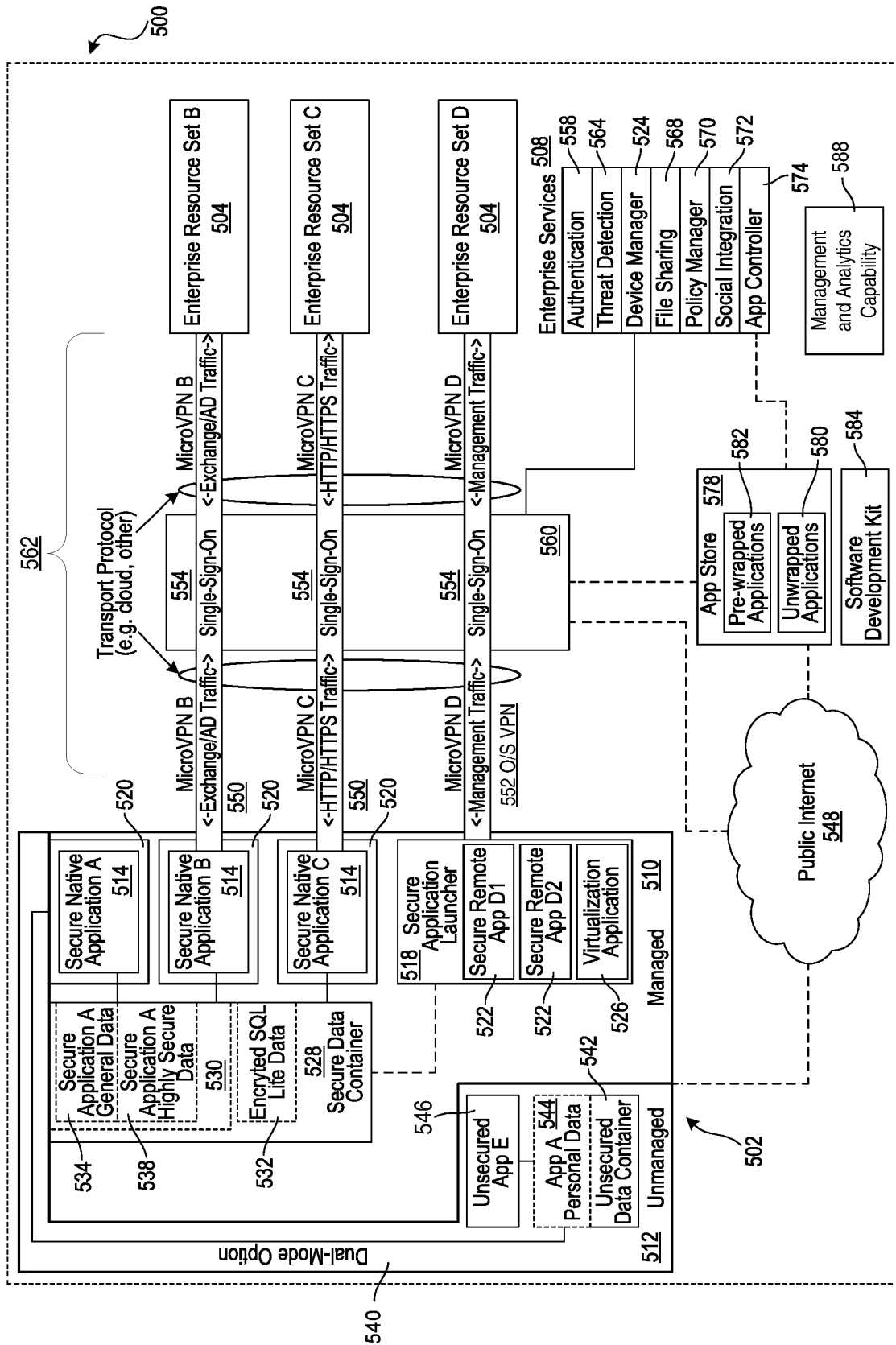


FIG. 5

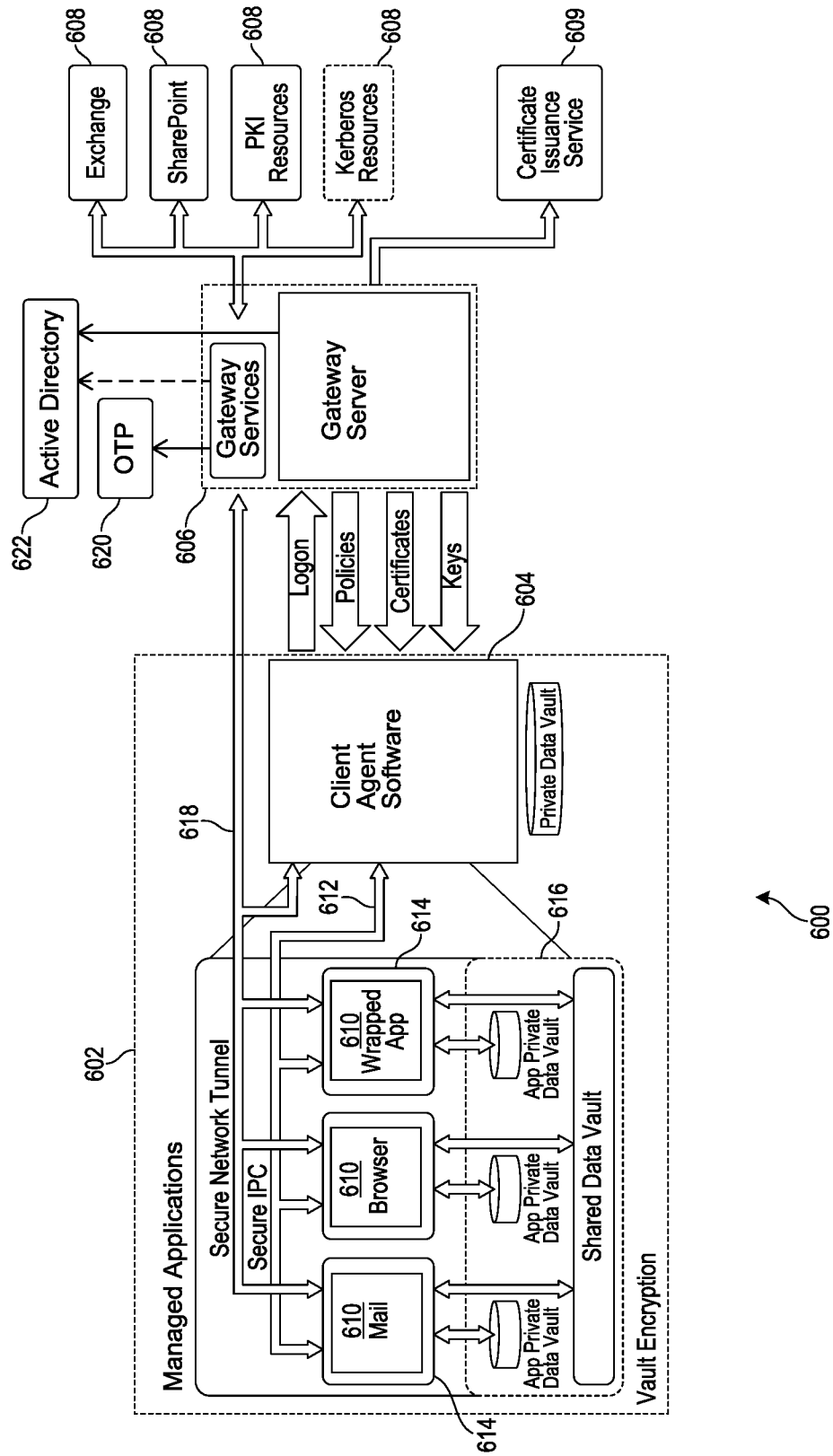


FIG. 6



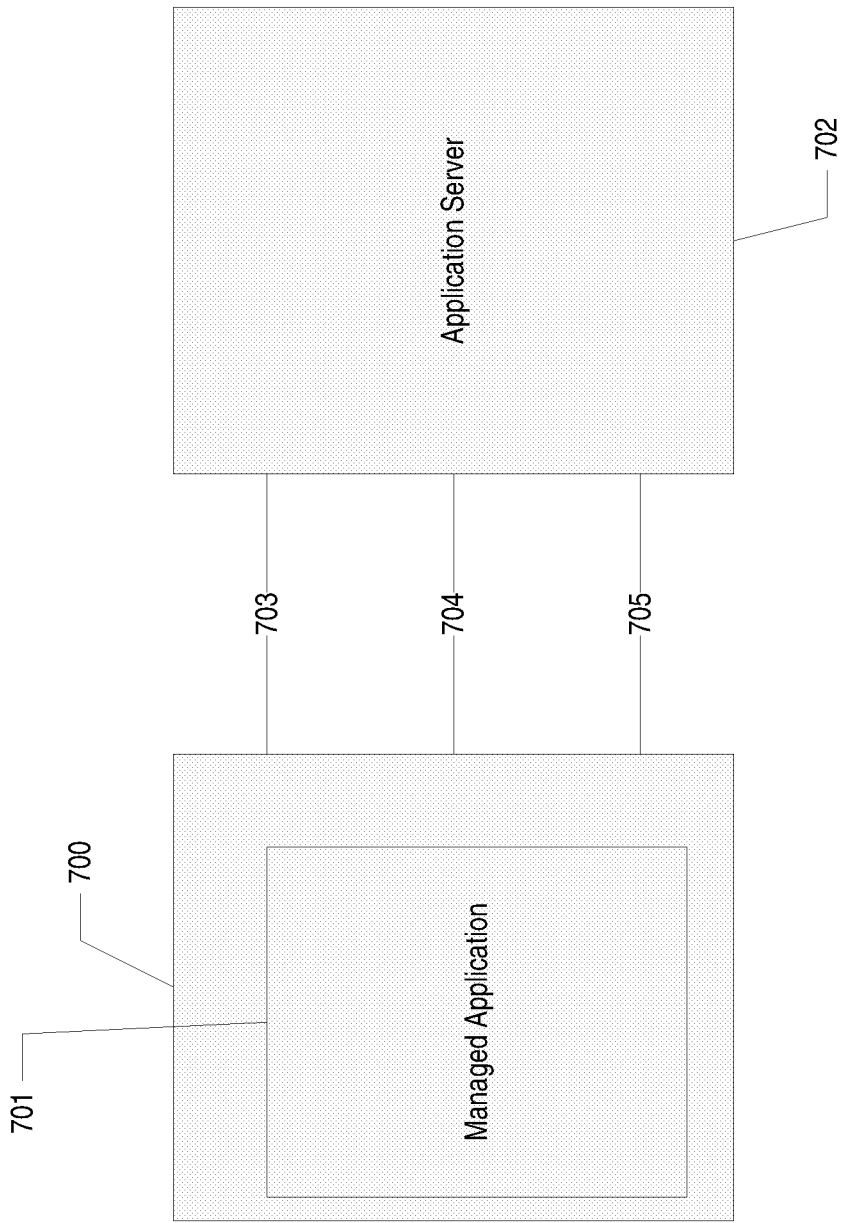


FIG. 7

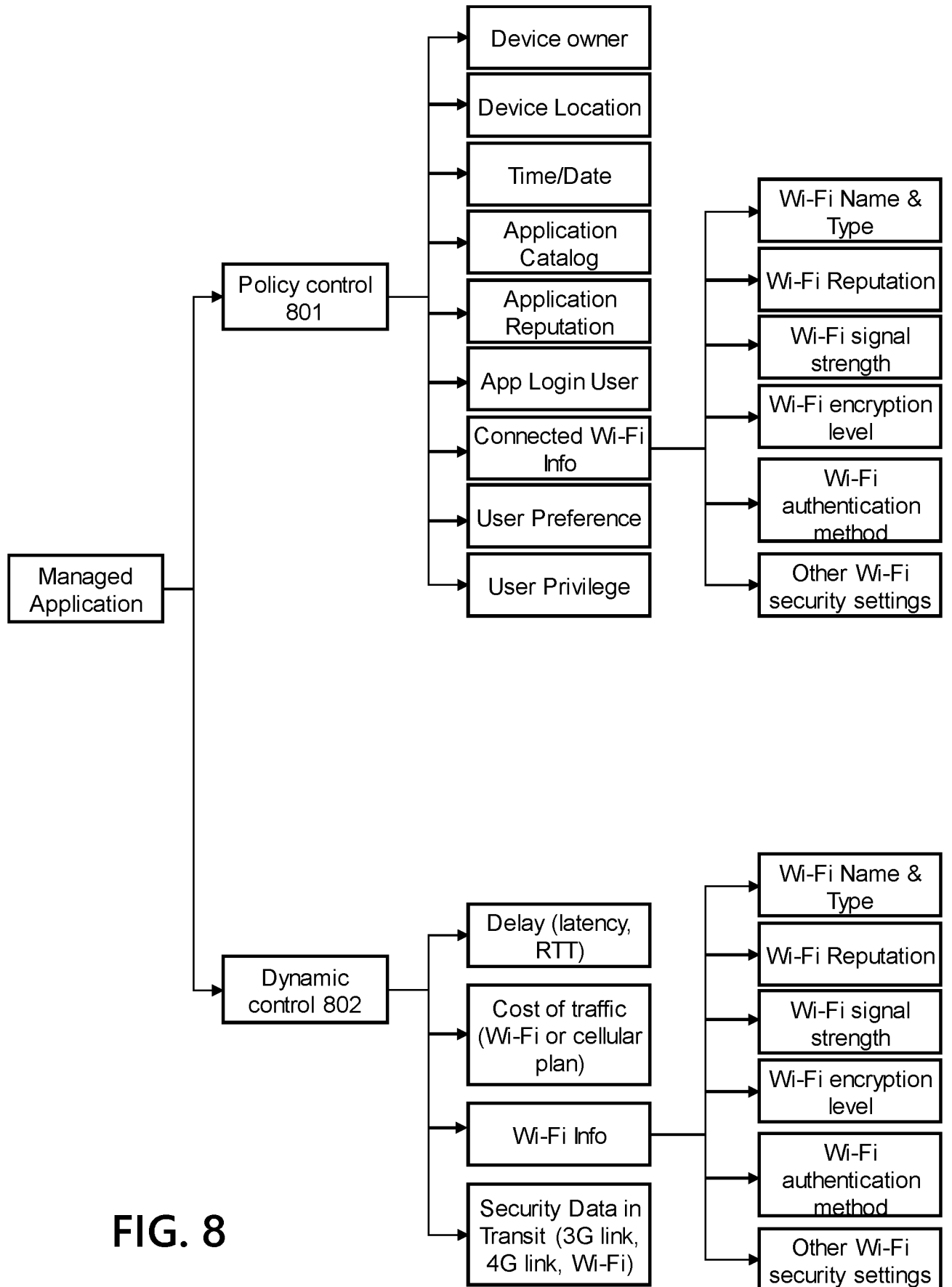


FIG. 8










Managed Application			SIM1 	SIM2 
	Social media	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Phone	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Web Browser	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Phone contacts	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	Backup Transport	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Backup Transport	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

FIG. 9



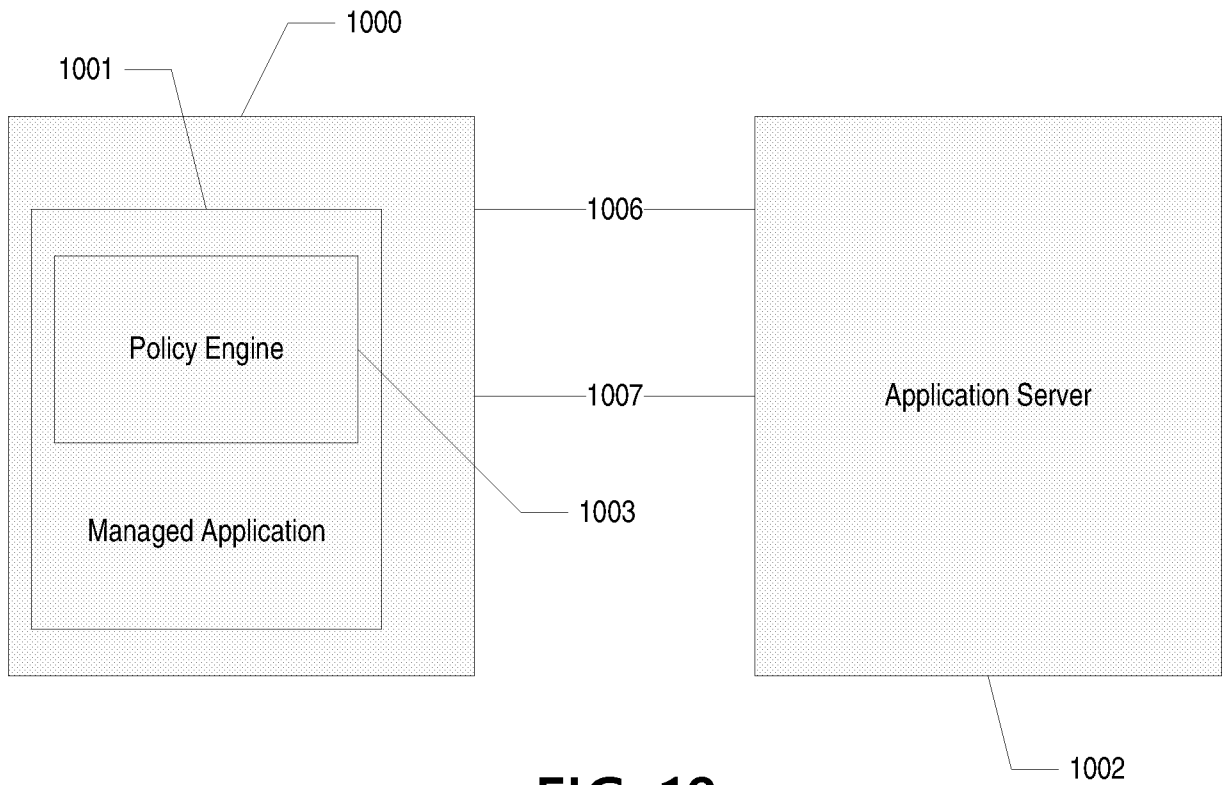


FIG. 10

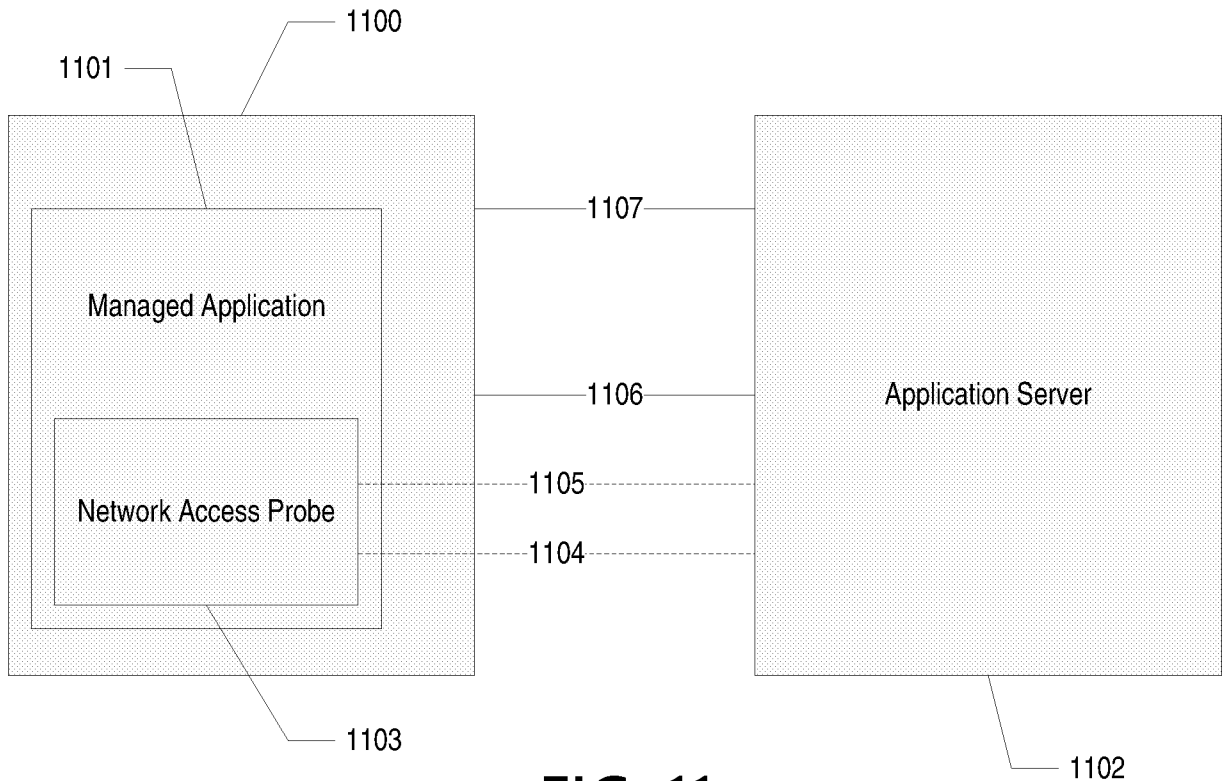


FIG. 11



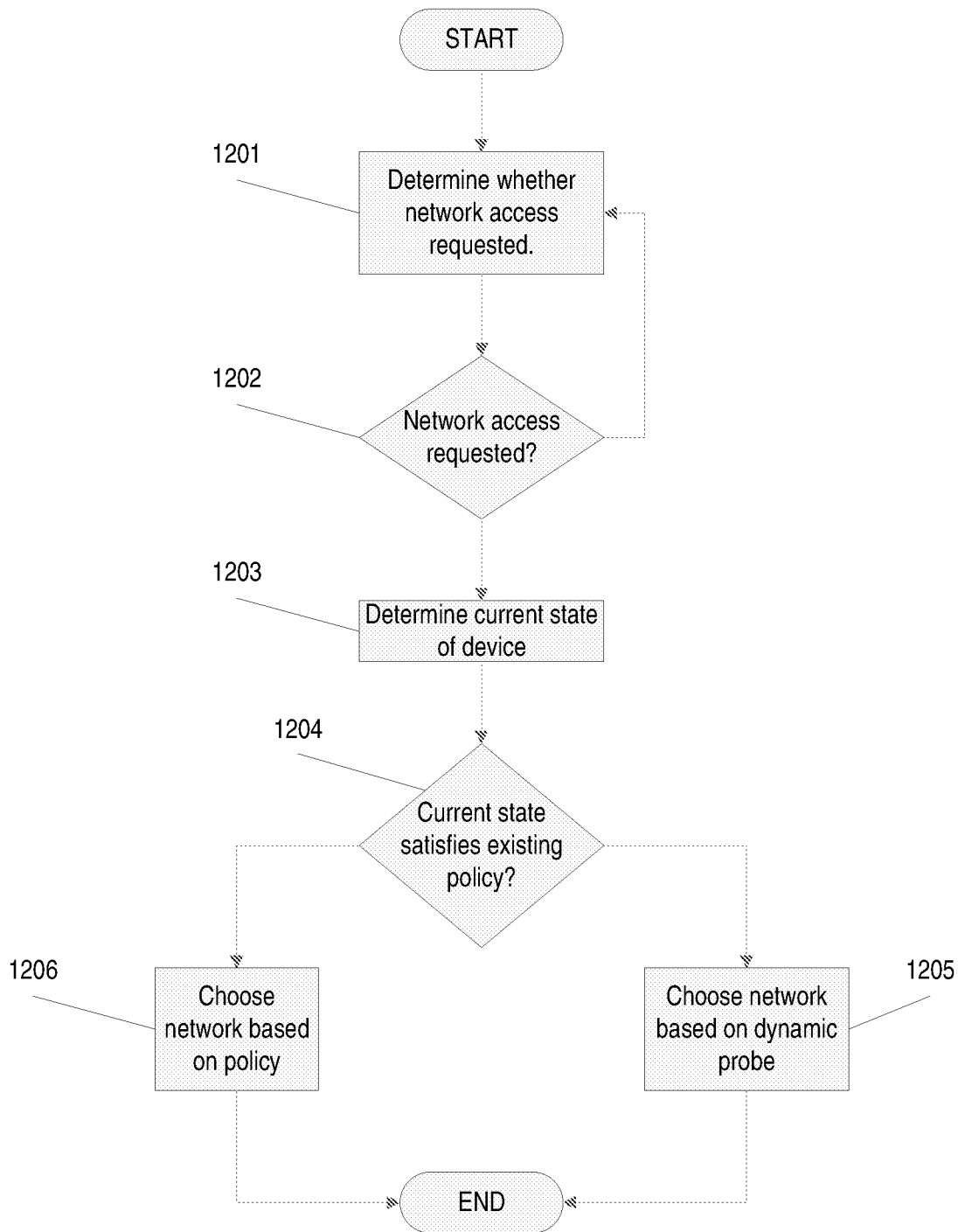


FIG. 12



INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2019/073032

A. CLASSIFICATION OF SUBJECT MATTER		
H04L 29/06(2006.01)i		
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) H04W; H04Q; H04M; H04L		
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) CNABS,CNKI,CNTXT,VEN,EPTXT,USTXT,WOTXT,3GPP:receiving,wireless,communication,transmit,recipient,channel, state, existing,policy, executing, engine, dynamic,probe, login, level, name,type		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2005255838 A1 (ADAMS NEIL PET AL.) 17 November 2005 (2005-11-17) the description, paragraphs [0002]-[0060] and claims 1-20	1-20
A	US 2006048155 A1 (WU YUH-CHERNGET AL.) 02 March 2006 (2006-03-02) the whole document	1-20
A	US 2013290426 A1 (SKY SOCKET LLC) 31 October 2013 (2013-10-31) the whole document	1-20
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> 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 19 September 2019		Date of mailing of the international search report 30 September 2019
Name and mailing address of the ISA/CN National Intellectual Property Administration, PRC 6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088 China		Authorized officer LU,Peng
Facsimile No. (86-10)62019451		Telephone No. 86-010-62411372

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2019/073032

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2005255838	A1	17 November 2005	DE	602005026643	D1	14 April 2011
				AU	2005239005	A1	10 November 2005
				WO	2005107144	A1	10 November 2005
				US	8005469	B2	23 August 2011
				AU	2009202857	B2	08 March 2012
				AT	500698	T	15 March 2011
				EP	1741225	B1	02 March 2011
				CN	1951060	A	18 April 2007
				CA	2564914	C	20 September 2016
				CN	1951060	B	30 November 2011
				KR	20070007198	A	12 January 2007
				US	RE46083	E1	26 July 2016
				US	RE46083	E	26 July 2016
				JP	4465387	B2	19 May 2010
				AU	2009202857	A1	06 August 2009
				BR	PI0510378	A	06 November 2007
				BR	PI0510378	B1	11 December 2018
				US	2010242086	A1	23 September 2010
				EP	1741225	A1	10 January 2007
				CN	102355466	A	15 February 2012
				HK	1167532	A1	04 November 2016
				JP	2007535247	A	29 November 2007
				HK	1099864	A1	29 July 2011
				KR	100926804	B1	12 November 2009
				US	RE44746	E1	04 February 2014
				CN	102355466	B	20 January 2016
				US	RE44746	E	04 February 2014
				US	7734284	B2	08 June 2010
				EP	1741225	A4	17 October 2007
				CA	2564914	A1	10 November 2005
<hr/>							
US	2006048155	A1	02 March 2006	US	7721288	B2	18 May 2010
<hr/>							
US	2013290426	A1	31 October 2013	US	2016154975	A1	02 June 2016
<hr/>							
				US	9270777	B2	23 February 2016
<hr/>							