



- (51) International Patent Classification:
G06F 21/24 (2006.01)
- (21) International Application Number:
PCT/US2012/030130
- (22) International Filing Date:
22 March 2012 (22.03.2012)
- (25) Filing Language:
English
- (26) Publication Language:
English
- (30) Priority Data:
13/069,271 22 March 2011 (22.03.2011) US
- (71) Applicant (for all designated States except US):
AMAZON TECHNOLOGIES, INC. [US/US]; P.O. Box 8102, Reno, Nevada 89507 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **BROOKER, Marc J.** [ZA/ZA]; 410 Terry Ave North, Seattle, Washington 98109-5210 (US). **BROWN, David** [ZA/ZA]; 410 Terry Ave North, Seattle, Washington 98109-5210 (US). **DE KADT, Christopher Richard Jaques** [ZA/ZA]; 410 Terry Ave North, Seattle, Washington 98109-5210 (US).
- (74) Agents: **ADAMS, Scott** et al.; Kilpatrick Townsend & Stockton LLP, Two Embarcadero Center, 8th Floor, San Francisco, California 94111 (US).

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

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

Declarations under Rule 4.17:

— as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

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

(54) Title: STRONG RIGHTS MANAGEMENT FOR COMPUTING APPLICATION FUNCTIONALITY

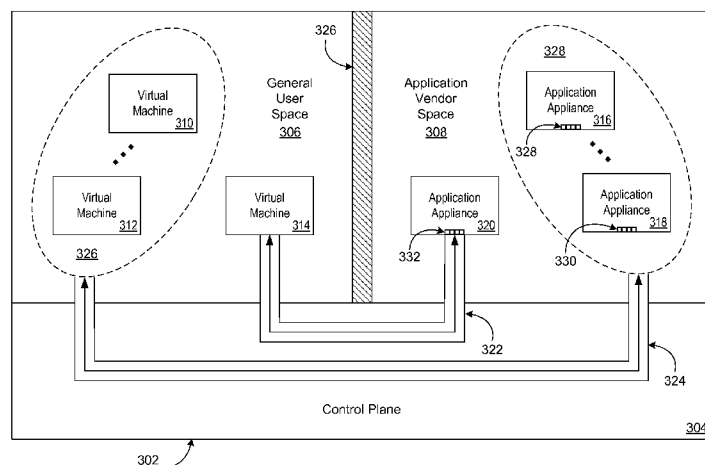
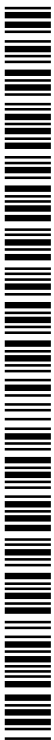


Figure 3

(57) Abstract: Illegal, unauthorized, uncompensated and/or under-compensated utilization of computing application functionality may be mitigated at least in part by controlling access to executable instructions implementing the computing application functionality. The executable instructions may be executed by a set of one or more virtual machines provisioned by a multi-tenant virtual resource provider. The virtual resource provider may provision the virtual machines and other virtual resources with a set of implementation resources managed by a control plane of the virtual resource provider. The control plane may perform a number of control functions for the virtual resource provider including management and enforcement of virtual resource access policies such as one or more policies collectively specifying that the computing application functionality is to be accessed in accordance with a license or agreement between a third party provider or vendor of the computing application functionality and a user of the computing application functionality.



STRONG RIGHTS MANAGEMENT FOR COMPUTING APPLICATION FUNCTIONALITY

BACKGROUND

5 **[0001]** From data processing and engineering to education and entertainment, computing devices have found a wide variety of applications in modern homes, schools and workplaces. Many such computing devices include processors capable of executing instructions (e.g., instructions corresponding to elements of a computer programming language), and much of the functionality of a computing device may be controlled by a set of executable instructions
10 and, optionally, a set of configuration data (e.g., by a computer program). Development of a computer program for a particular application and/or set of functionality can require a significant investment of time and resources. For example, years of effort by teams of dozens of people is not uncommon. However, executable instructions and configuration data can have a digital representation (e.g., an application “executable” or “binary”) that is easily
15 copied, and illegal and/or uncompensated use of enabled functionality (e.g., application “piracy”) is a significant problem.

[0002] Several conventional “rights management” schemes (e.g., “copy-protection” schemes) attempt to address such illegal and/or uncompensated use. For example, some conventional rights management schemes involve cryptographic keys that unlock
20 corresponding sets of application functionality. Some conventional rights management schemes involve authentication and/or periodic re-authentication with a remote server (e.g., remote in a communication network). Some conventional rights management schemes involve checking for the local presence of a physical computing device component (e.g., a “dongle”). However, conventional rights management schemes have disadvantages.

25 **[0003]** For example, the copy enabling the illegal and/or uncompensated use of application functionality may control and/or be installed on computing device hardware to which the user has physical access. Even where portions of executable instructions and/or

configuration data begin encrypted and/or locked, such physical access can enable the user to obtain corresponding decrypted and/or unlocked portions, or otherwise circumvent the need to obtain a legitimate key. Such physical access may also enable the user to emulate, or otherwise circumvent the need for, a remote authentication server and/or a local dongle.

5 Remote access to low-level computing device functionality (e.g., operating system-level functionality) and/or access to low-level functionality of a communication network connected to the computing device (e.g., access to in-transit data packet “sniffing”) can similarly enable a user intent on illegal and/or uncompensated use of application functionality.

10

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] Various embodiments in accordance with the present disclosure will be described with reference to the drawings, in which:

[0005] Figure 1 is a schematic diagram illustrating an example environment for
15 implementing aspects in accordance with at least one embodiment;

[0006] Figure 2 is a schematic diagram depicting aspects of an example virtual resource provisioning architecture in accordance with at least one embodiment;

[0007] Figure 3 is a schematic diagram depicting aspects of an example virtual resource provider in accordance with at least one embodiment;

20 **[0008]** Figure 4 is a schematic diagram depicting aspects of an example control plane in accordance with at least one embodiment;

[0009] Figure 5 is a flowchart depicting example steps for making an application appliance available at a virtual resource provider in accordance with at least one embodiment; and

[0010] Figure 6 is a flowchart depicting example steps for accessing application appliance
25 functionality in accordance with at least one embodiment; and

[0011] Figure 7 is a flowchart depicting example steps for dynamic feature activation in accordance with at least one embodiment; and

[0012] Figure 8 is a flowchart depicting example steps for workflow management in accordance with at least one embodiment.

[0013] Same numbers are used throughout the disclosure and figures to reference like components and features, but such repetition of number is for purposes of simplicity of explanation and understanding, and should not be viewed as a limitation on the various
5 embodiments.

DETAILED DESCRIPTION

[0014] In the following description, various embodiments will be described. For purposes
10 of explanation, specific configurations and details are set forth in order to provide a thorough understanding of the embodiments. However, it will also be apparent to one skilled in the art that the embodiments may be practiced without the specific details. Furthermore, well-known features may be omitted or simplified in order not to obscure the embodiment being described.

[0015] In at least one embodiment, illegal, unauthorized, uncompensated and/or under-
15 compensated utilization of computing application functionality may be mitigated at least in part by controlling access to executable instructions that implement the computing application functionality. The executable instructions may be executed by a set of one or more virtual computing machines (“virtual machines”) provisioned by a multi-tenant virtual
20 resource provider. The virtual resource provider may provision the virtual machines and other virtual resources with a managed set of implementation resources such as physical servers, physical network switches and physical network paths. The provisioning, including allocation and ongoing reallocation of the implementation resources, may be managed by a control plane of the virtual resource provider. The control plane may perform a number of
25 control functions for the virtual resource provider including management and enforcement of virtual resource access policies.

[0016] For example, the virtual resource provider may provision the set of virtual machines and a set of communication connections enabling communication with the set of virtual machines. The set of virtual resource access policies enforced by the control plane of the

virtual resource provider may include one or more policies collectively specifying that the provisioned set of virtual machines executing the executable instructions that implement the computing application functionality are to be accessed with the provisioned set of communication connections (the “allowed” set of communication connections), and no others. Where a communication protocol allows specification of a communication port or a sub-address or the like, such policies may specify the allowed communication connections to a finest level of granularity. The set of virtual resource access policies may further include one or more policies collectively specifying that the computing application functionality is to be accessed in accordance with a license or agreement between a third party provider or vendor of the computing application functionality and a user of the computing application functionality.

[0017] In at least one embodiment, the allowed set of communication connections corresponds to communication connections between virtual machines provisioned by the virtual resource provider. For example, the allowed set of communication connections may be between the provisioned set of virtual machines executing the executable instructions that implement the computing application functionality (the “application appliance”) and one or more virtual machines provisioned by the virtual resource provider at which a user account and work environment is maintained by an operating system (one or more “user VMs”). In at least one embodiment, the allowed set of communication connections may include communication connections between the application appliance and one or more virtual machines and/or computing devices not provisioned by the virtual resource provider, and participating in a virtual private computing cloud (VPC) maintained by the virtual resource provider such that the control plane may enforce access policies with respect to the application appliance and/or the allowed set of communication connections.

[0018] Various approaches may be implemented in various environments for various applications. For example, Figure 1 illustrates aspects of an example environment 100 for implementing aspects in accordance with various embodiments. As will be appreciated, although a Web-based environment may be utilized for purposes of explanation, different environments may be utilized, as appropriate, to implement various embodiments. The environment 100 shown includes both a testing or a development portion (or side) and a

production portion. The production portion includes an electronic client device 102, which may include any appropriate device operable to send and receive requests, messages, or information over an appropriate network 104 and convey information back to a user of the device 102. Examples of such client devices include personal computers, cell phones, handheld messaging devices, laptop computers, tablet computers, set-top boxes, personal data assistants, electronic book readers, and the like.

[0019] The network 104 may include any appropriate network, including an intranet, the Internet, a cellular network, a local area network, a wide area network, a wireless data network, or any other such network or combination thereof. Components utilized for such a system may depend at least in part upon the type of network and/or environment selected. Protocols and components for communicating via such a network are well known and will not be discussed herein in detail. Communication over the network may be enabled by wired or wireless connections, and combinations thereof. In this example, the network 104 includes the Internet, as the environment includes a Web server 106 for receiving requests and serving content in response thereto, although for other networks an alternative device serving a similar purpose could be utilized as would be apparent to one of ordinary skill in the art.

[0020] The illustrative environment 100 includes at least one application server 108 and a data store 110. It should be understood that there may be several application servers, layers, or other elements, processes, or components, which may be chained or otherwise configured, which may interact to perform tasks such as obtaining data from an appropriate data store. As used herein the term “data store” refers to any device or combination of devices capable of storing, accessing, and/or retrieving data, which may include any combination and number of data servers, databases, data storage devices, and data storage media, in any standard, distributed, or clustered environment.

[0021] The application server 108 may include any appropriate hardware and software for integrating with the data store as needed to execute aspects of one or more applications for the client device 102, and may even handle a majority of the data access and business logic for an application. The application server 108 provides access control services in cooperation with the data store 110, and is able to generate content such as text, graphics,

audio, and/or video to be transferred to the user, which may be served to the user by the Web server 106 in the form of HTML, XML, or another appropriate structured language in this example.

[0022] The handling of all requests and responses, as well as the delivery of content
5 between the client device 102 and the application server 108, may be handled by the Web server 106. It should be understood that the Web and application servers 106, 108 are not required and are merely example components, as structured code discussed herein may be executed on any appropriate device or host machine as discussed elsewhere herein. Further, the environment 100 may be architected in such a way that a test automation framework may
10 be provided as a service to which a user or application may subscribe. A test automation framework may be provided as an implementation of any of the various testing patterns discussed herein, although various other implementations may be utilized as well, as discussed or suggested herein.

[0023] The environment 100 may also include a development and/or testing side, which
15 includes a user device 118 allowing a user such as a developer, data administrator, or tester to access the system. The user device 118 may be any appropriate device or machine, such as is described above with respect to the client device 102. The environment 100 may also include a development server 120, which functions similar to the application server 108 but typically runs code during development and testing before the code is deployed and executed
20 on the production side and becomes accessible to outside users, for example. In some embodiments, an application server may function as a development server, and separate production and testing storage may not be utilized.

[0024] The data store 110 may include several separate data tables, databases, or other data storage mechanisms and media for storing data relating to a particular aspect. For example,
25 the data store 110 illustrated includes mechanisms for storing production data 112 and user information 116, which may be utilized to serve content for the production side. The data store 110 also is shown to include a mechanism for storing testing data 114, which may be utilized with the user information for the testing side. It should be understood that there may be many other aspects that are stored in the data store 110, such as for page image

information and access right information, which may be stored in any of the above listed mechanisms as appropriate or in additional mechanisms in the data store 110.

[0025] The data store 110 is operable, through logic associated therewith, to receive instructions from the application server 108 or development server 120, and obtain, update, or otherwise process data in response thereto. In one example, a user might submit a search request for a certain type of item. In this case, the data store 110 might access the user information 116 to verify the identity of the user, and may access the catalog detail information to obtain information about items of that type. The information then may be returned to the user, such as in a results listing on a Web page that the user is able to view via a browser on the user device 102. Information for a particular item of interest may be viewed in a dedicated page or window of the browser.

[0026] Each server typically will include an operating system that provides executable program instructions for the general administration and operation of that server, and typically will include a computer-readable medium storing instructions that, when executed by a processor of the server, allow the server to perform its intended functions. Suitable implementations for the operating system and general functionality of the servers are known or commercially available, and are readily implemented by persons having ordinary skill in the art, particularly in light of the disclosure herein.

[0027] The environment 100 in one embodiment is a distributed computing environment utilizing several computer systems and components that are interconnected via communication links, using one or more computer networks or direct connections. However, it will be appreciated by those of ordinary skill in the art that such a system could operate equally well in a system having fewer or a greater number of components than are illustrated in Figure 1. Thus, the depiction of the system 100 in Figure 1 should be taken as being illustrative in nature, and not limiting to the scope of the disclosure.

[0028] In at least one embodiment, one or more aspects of the environment 100 may incorporate and/or be incorporated into a virtual resource provisioning architecture. Figure 2 depicts aspects of an example virtual resource provisioning architecture 200 in accordance with at least one embodiment. The example virtual resource provisioning architecture 200 includes multiple clients 202-204 communicatively connected to a virtual resource provider

206 over a network 208. For example, the clients 202-204 may correspond to computing devices such as the computing device 102 of Figure 1 and/or client programs incorporated into such computing devices. The ellipsis between the client 202 and the client 204 indicates that the virtual resource provisioning architecture 200 may include any suitable number of clients although, for clarity, only two are shown in Figure 2. Ellipses are used similarly throughout the drawings.

[0029] One or more of the clients 202-204 may be utilized by one or more authorized users associated with a tenant of the virtual resource provider 206 to interact with a control plane 210 of the virtual resource provider 206, and thereby provision one or more virtual computing resources 212. Alternatively, or in addition, one or more of the clients 202-204 may be utilized to interact with provisioned virtual computing resources 212. The provisioned virtual computing resources 212 may include any suitable type and/or number of virtual resources 214-216. Examples of suitable virtual resources 214-216 include virtual machines such as virtual computer systems (VCSs), virtual networks, virtual private networks (VPNs), virtual network connections, virtual data stores, virtual file system volumes, specialized data processing agents, media streaming agents including audio and video streaming agents, message queues, publish-subscribe topics configured to notify subscribers having subscriptions that match events published to the publish-subscribe topics, monitoring agents, load balancing agents, and suitable combinations thereof.

[0030] The virtual resource provider 206 may further include any suitable type and/or number of implementation resources 218. Each of the provisioned computing resources 212 may be implemented by a set of the implementation resources 218. In at least one embodiment, various implementation resources of the implementation resources 218 may be configured to participate in implementing, at least in part, multiple of the provisioned computing resources 212. Examples of suitable implementation resources 218 include VCS servers, data store servers, computers, server racks, networking hardware including switches, routers, gateways, bridges, hubs, repeaters, firewalls and wireless transceivers, power supplies, generators, data centers, rooms in data centers, mobile data centers, as well as non-volatile storage devices including hard drives, processing units such as central processing units (CPUs), caches in processing units, processing cores in multi-core processing units,

volatile storage devices such as memory modules including random access memory (RAM) modules, and RAM chips of multi-chip memory modules, network interface hardware and suitable combinations thereof.

[0031] In at least one embodiment, one or more types of provisioned computing resource 212, such as virtual computer systems, are implemented by default with a set of implementation resources having a standardized set of implementation resource capacities (e.g., a standardized amount of volatile and/or non-volatile storage). Different implementation resource capacities may be provisioned for such computing resources 212. For example, such computing resources 212 may be provisioned with implementation resources collectively having a set of implementation resource capacities one or more of which is a multiple of a corresponding implementation resource capacity in the standardized set. Suppose a virtual computer system with 1 gigabyte of available RAM corresponds to a “small” size. Virtual computer systems with “medium” and “large” sizes, corresponding to 2 gigabytes and 4 gigabytes of RAM, respectively, may be requested, for example. Provisioned computing resources 212 with larger “sizes” may have commensurately higher associated costs.

[0032] The provisioned virtual computing resources 212 may further include any suitable type and/or number of application appliances 220-222. In at least one embodiment, an application appliance may configure a set of one or more virtual resources (e.g., corresponding to the virtual resources 214-216) and/or the implementation resources 218 to provide a set of computing application functionality. Application appliances 220-222 may be provisioned in a manner corresponding to that of provisioning the virtual resources 214-216. In the example virtual resource provider 206, application appliances 220-222 are located in an application vendor space 224 distinct from a general user space 226 of the provisioned computing resources 212. Location in different provisioned computing resource spaces 224-226 may correspond to different access policy and/or cost accounting treatments reflecting different roles with respect to the virtual resource provider 206.

[0033] For example, virtual resources 214-216 provisioned in the general user space 226 may facilitate a business end-use of a tenant. In contrast, application appliances 220-222 may be offered by third party vendors to provide a set of computing application

functionality. Access policies associated with virtual resources 214-216 in the general user space 226 may allow access from public networks. In contrast, access policies associated with application appliances 220-222 may restrict access to other provisioned computing resources 212 or to a particular subset of the virtual resources 214-216 such as a particular set of user VMs and/or communication connections. Costs associated with virtual resources 214-216 in the general user space 226 may be determined based at least in part on allocated implementation resources 218. In contrast, costs associated with application appliances 220-222 may be determined based at least in part on a flat fee, a fee per suitable unit of time, associated implementation resource 218 costs plus a surcharge, feature usage, and/or any suitable cost accounting method.

[0034] The control plane 210 may provision computing resources 212 with implementation resources 218 responsive to provisioning requests. The control plane 210 may further manage and enforce policies that control access to the provisioned computing resources, including one or more policies that define and/or maintain the application vendor space 224 distinct from the general user space 226. The control plane 210 may further track costs associated with maintaining the provisioned computing resources 212 and allocate the costs as appropriate to tenant accounts. An example control plane in accordance with at least one embodiment is described below in more detail with reference to Figure 4.

[0035] In at least one embodiment, access to executable instructions that implement the computing application functionality of an application appliance 220-222 is controlled at least in part by enforcing at least one policy specifying that particular application appliances 220-222 be accessed through a particular set of communication connections, and no other. Figure 3 depicts an example virtual resource provider 302 in accordance with at least one embodiment. The example virtual resource provider 302 of Figure 3 includes a control plane 304, a general user space 306 and an application vendor space 308 corresponding to the control plane 210, the general user space 226 and the application vendor space 224 of Figure 2. The general user space 306 of Figure 3 contains multiple virtual machines 310-314 communicatively connected to multiple application appliances 316-320 with multiple provisioned communication connections 322-324. For example, the virtual machines 310-314 may be user VMs, and the application appliances 316-320 may correspond to the

application appliances 220-222 of Figure 2. In at least one embodiment, user control over application appliances 316-320 is at a reduced level relative to the virtual machines 310-314. For example, user control over the application appliances 316-320 may be limited to starting, suspending and terminating the application appliances 316-320. In contrast, authorized users
5 may be able to comprehensively configure and login to the virtual machines 310-314.

[0036] In the example virtual resource provider 302, the general user space 306 and the application vendor space 308 are separated by a communicative barrier 326 to indicate that ad hoc and/or noncompliant communication connections between the virtual machines 310-314 and the application appliances 316-320 are prevented by one or more policies enforced
10 by the control plane 304. One or more of the virtual machines 310-314 may be connected to one or more of the application appliances 316-320 with policy-complaint communication connections 322-324. In the example virtual resource provider 302, the virtual machine 314 is connected to the application appliance 320 with policy-compliant communication connection 322. The set 326 of virtual machines 310-312 are connected to the set of
15 application appliances 316-318 with policy-compliant communication connection 324. The policy-compliant communication connections 322-324 are depicted as passing through the control plane 304 to indicate the ability of the control plane 304 to enforce associated access policies.

[0037] The policy-compliant communication connections 322-324 may be maintained with
20 any suitable communication media and/or communication protocol. For example, the policy-compliant communication connections 322-324 may be maintained with a communication protocol in accordance with a transmission control protocol and/or an internet protocol (e.g., TCP/IP). Each virtual machine 310-314 and/or application appliance 316-320 may be associated with a communication protocol address and/or communication port and, for
25 example, the access policy set associated with the communication connection 322 may specify that a destination of protocol messages conveyed through the communication connection 322 correspond to a particular communication protocol address and a particular communication port. Alternatively, or in addition, the application appliances 316-320 may incorporate and/or provide one or more interfaces 328-332 to the computing application
30 functionality, and, for example, the access policy set may specify that protocol messages

conveyed through the communication connection 322 be in accordance with and/or directed to one or more elements of the interface 332 (e.g., a selected subset of such interface elements).

[0038] The interfaces 328-332 may include any suitable interface elements such as interface elements corresponding to functionality, or sets of functionality, of the computing application. The interfaces 328-332 may incorporate and/or be incorporated in a user interface (UI) such as a graphical user interface (GUI), a Web-based interface, a programmatic interface such as an application programming interface (API) and/or a set of remote procedure calls (RPCs) corresponding to provisioning interface elements, a messaging interface such as a messaging interface in which the interface elements of the interfaces 328-332 correspond to messages of a communication protocol, a remote desktop protocol such as a remote framebuffer protocol (e.g., RFB) or an "X WINDOW SYSTEM" protocol as described in Scheifler et al., "The X Window System," ACM Transactions on Graphics, April 1986, pages 79-109, and/or any suitable combination thereof. Web-based interfaces may include Web services interfaces such as Representational State Transfer (REST) compliant ("RESTful") Web services interfaces or Simple Object Access Protocol (SOAP) compliant Web services interfaces or other "non-RESTful" Web services interfaces.

[0039] Figure 4 depicts aspects of an example control plane 402 in accordance with at least one embodiment. The control plane 402 may include a user interface (I/F) 404 enabling authorized users to access control plane 402 functionality, and an application vendor interface (I/F) 406 enabling an application vendor to manage a set of application appliances (e.g., application appliances 316-320 of Figure 3) offered by the application vendor. The user interface 404 and the application vendor interface 406 may incorporate and/or be incorporated in any suitable type of functionality interface (e.g., as described for interfaces 328-332 of Figure 3).

[0040] The virtual resource provider 302 (Figure 3) incorporating the control plane 402 may have multiple tenants responsible for costs associated with computing resources 212 (Figure 2) provisioned by tenant-authorized users. An administrative user designated by a tenant may interact with the user interface 404 to manage different types of users associated with the tenant, including users authorized to incur costs, for example, by provisioning

computing resources 212. Authorized users may interact with the user interface 404 to provision computing resources 212, and manage (e.g., view, label, allocate, route and discharge) associated costs.

[0041] An application vendor may also be a tenant of the virtual resource provider 302 (Figure 3), although this is not necessary in each embodiment. The application vendor may interact with the application vendor interface 406 to configure and/or register application appliances (such as the application appliances 316-320 of Figure 3) as available for provisioning, as well as specify license conditions, configure associated cost plans and manage associated costs. The license conditions may include any suitable conditions with respect to access of the computing application functionality such as that a valid and unexpired license exist, that no more than a maximum number of users has accessed the computing application functionality or some specified portion thereof, that no more than a maximum number of concurrent users is accessing the computing application functionality or some specified portion thereof, that the computing application functionality or some specified portion thereof has been accessed no more than a threshold number of times, and the like. Depending on the associated cost plan, the application vendor may be responsible to the virtual resource provider 302 for costs incurred by provisioned instances of application appliances offered by the application vendor. Alternatively, or in addition, the associated cost plan may specify that the provisioning tenant is responsible for associated costs, and fees paid by the provisioning tenant may be allocated between the application vendor and the virtual resource provider in accordance with an agreement between them.

[0042] A provisioning component 408 of the control plane 402 may provision computing resources 212 (Figure 2) responsive to provisioning requests, for example, received from the user interface 404. The provisioning component 408 may determine types and capacities of implementation resources 218 required to implement particular provisioned computing resources 212 and allocate available such implementation resources to the task of implementing virtual resources 214-216 and/or application appliances 220-222, as well as ongoing re-allocation of implementation resources 212, for example, to increase utilization efficiency and/or to lower a chance of provisioned resource failure due to implementation resource failure.

[0043] A policy enforcement component 410 of the control plane 402 may manage and enforce virtual resource provider 206 (Figure 2) policies. For example, the policy enforcement component 410 may receive policies to be enforced from an authorized user through the user interface 404, policies with respect to a particular provisioned resource may be established at the policy enforcement component 410 during provisioning, policies may be established at the policy enforcement component 410 by an administrator of the virtual resource provider 206, and/or policies (e.g., cryptographically signed policies) may be received along with provisioned resource 212 access and/or interaction requests from clients 202-204. Virtual resource provider 206 policies may govern any suitable aspect of virtual resource provider 206 functionality including functionality provided by provisioned resources 212. Particular sets and/or subsets of functionality provided by provisioned resources 212 may be named, labeled and/or addressable. Each such set and/or subset may be individually governed with virtual resource provider 206 policies. Such governance may include constraint with respect to implementation resource allocation and utilization, as well as access by users and transfer of data to and from particular provisioned resources 212. Users of provisioned resources 212 may include client 202-204 users including anonymous users, virtual resource provider 206 users including administrative users, and virtual resource provider 206 components including implementation resources 218, provisioned resources 212, and control plane 402 components 404-416.

[0044] A virtual resource provider 206 (Figure 2) policy may specify any suitable set of conditions to be satisfied. For example, the policy may specify conditions under which access to a particular application appliance is permitted. Such conditions may be specified with any suitable condition specification language including suitable programming languages, and may include compound conditions, for example, specified with Boolean operators. Condition parameters may include any suitable data available to the virtual resource provider 206. Condition parameter examples include environmental data such as calendar date and time of day, and request-associated data such as originating network address, originating geographical location, originating political and/or administrative division and communication protocol employed.

[0045] A cost-tracking component 412 of the control plane 402 may track costs (e.g., computation and/or financial costs) associated with provisioning and/or maintaining the computing resources 212 (Figure 2). Costs may be allocated to accounts including tenant accounts. For example, costs associated with computing resources 212 provisioned by one or more users associated with a particular tenant may be allocated to the tenant's account. A tenant account and/or one or more of the provisioned resources 212 may be associated with one or more cost plans, and the costs allocated to the tenant account may be determined in accordance with the cost plan(s). A cost plan may specify costs as flat fees and/or based on any suitable metric. For example, the cost plan may specify costs based on a number of units of time that a particular provisioned resource 212 is available to at least one user associated with the tenant, a number of units of time that a particular implementation resource 218 is allocated to maintaining provisioned resources 212 associated with the tenant, a number of uses of a particular set of features of a particular provisioned resource 212, and/or suitable combinations thereof. With respect to application appliances 220-222, the cost plan may specify a cost accounting relationship with the tenant including cost pass-through, cost plus a surcharge, flat fee, periodic access fee, feature access fee, activation and deactivation fees, independent billing, and suitable combinations thereof.

[0046] An application rights management (ARM) component 414 of the control plane 402 may act to establish and maintain user and vendor rights with respect to provisioned application appliances 220-222 (Figure 2). For example, the application rights management component 414 may provide and/or establish virtual resource provider 206 policies that control access to executable instructions that implement functionality of the provisioned application appliances 220-222. The application rights management component 414 may further facilitate activation and/or deactivation of sets of application functionality and/or application features. For example, the application rights management component 414 may notify application appliances 220-222 of user requests to activate and/or deactivate application features, and modify virtual resource provider 206 policies and/or cost plans responsive to activation status updates received from application appliances 220-222.

[0047] The control plane 402 may further include a workflow component 416 configured at least to establish and maintain workflows such as provisioned resource workflows,

provisioning workflows and/or policy enforcement workflows established by provisioned resources 212 (Figure 2), the provisioning component 408 and the policy enforcement component 410, respectively. Workflows may include one or more sequences of tasks to be executed to perform a job, such as virtual resource configuration, provisioning or policy management. A workflow, as the term is used herein, is not the tasks themselves, but a task control structure that may control flow of information to and from tasks, as well as the order of execution of the tasks it controls. For example, a workflow may be considered a state machine that can manage and return the state of a process at any time during execution. Workflows may be created from workflow templates. For example, a policy enforcement workflow may be created from a policy enforcement workflow template configured with parameters by the policy enforcement component 410.

[0048] The workflow component 416 may modify, further specify and/or further configure established workflows. For example, the workflow component 416 may select particular implementation resources of the virtual resource provider 206 (Figure 2) to execute and/or be assigned to particular tasks. Such selection may be based at least in part on the computing resource needs of the particular task as assessed by the workflow component 416. As another example, the workflow component 416 may add additional and/or duplicate tasks to an established workflow and/or reconfigure information flow between tasks in the established workflow. Such modification of established workflows may be based at least in part on an execution efficiency analysis by the workflow component 416. For example, some tasks may be efficiently performed in parallel, while other tasks depend on the successful completion of previous tasks.

[0049] The control plane 402 may be implemented with a set of provisioned resources 212 (Figure 2), a set of implementation resources 218 and/or corresponding computing resources. Each of the implementation resources 218 may be controlled by the control plane 210. For example, each implementation resource may participate in and/or incorporate a portion, agent and/or component of the control plane 210. Each of the provisioned resources 212 may be controlled by the control plane 210. For example, each provisioned resource may participate in and/or incorporate a portion, agent and/or component of the control plane 210. The control plane 210 may be distributed throughout the implementation resources 218

and/or the provisioned resources 212. For example, the control plane 210 may be implemented with distributed computing techniques well known to those of skill in the art.

[0050] The description now turns to example steps that may be performed in accordance with at least one embodiment. Figure 5 depicts example steps for making an application appliance available at a virtual resource provider in accordance with at least one embodiment. At step 502, a prototype application appliance may be configured. An authorized user of a third party application vendor may provision a virtual machine at the virtual resource provider 206 (Figure 2) and configure the virtual machine to execute instructions that implement a desired set of computing application functionality. For example, the virtual machine may be a virtual computer system incorporating a computer operating system, and the authorized user may install and configure one or more application modules into the virtual computer system and/or the computer operating system. Alternatively, the virtual machine may incorporate the desired set of computing application functionality independent of a computer operating system.

[0051] At step 504, the prototype application appliance may be packaged into a form suitable for provisioning. For example, the authorized user may request that the virtual resource provider 206 (Figure 2) create the provisionable package from the prototype configured at step 502. The user interface 404 and/or the application vendor interface 406 (Figure 4) may include one or more interface elements enabling the authorized user to make such requests. At step 506, the packaged prototype may be submitted to and/or registered with the virtual resource provider 206. For example, the application vendor interface 406 may include one or more interface elements enabling such submissions and/or registrations. Step 506 may be incorporated into step 504.

[0052] At step 508, one or more application appliance feature costs may be specified. For example, the authorized user may interact with one or more interface elements of the application vendor interface 406 (Figure 4) to specify a cost plan for users of the application appliance. Costs associated with access to basic features may be specified, as well as costs associated with each of a set of non-basic and/or premium features. Application-specific feature codes may be associated with human-readable names, short descriptions and/or long descriptions. At step 510, a request may be made to make the application appliance available

for provisioning. For example, the authorized user may submit the request with one or more interface elements of the application vendor interface 406.

[0053] At step 512, the submitted and/or registered application appliance prototype may be verified. For example, the application rights management component 414 (Figure 4) may verify a static and/or dynamic integrity of the application appliance prototype including with respect to security. If the application appliance is verified, then at step 516 it may be made available for provisioning by authorized users of tenants of the virtual resource provider 206 (Figure 2). Otherwise, one or more problems that occurred during verification may be reported to the vendor at step 514.

[0054] Figure 6 depicts example steps for accessing application appliance functionality in accordance with at least one embodiment. At step 602, a request to provision a user VM may be received. For example, an authorized user associated with a tenant of the virtual resource provider 206 (Figure 2) may submit a provisioning request with the user interface component 404 (Figure 4) of the control plane 402. At step 604, the requested user VM may be provisioned. For example, the provisioning component 408 may provision the requested virtual machine 314 in the general user space 306 (Figure 3). The requested virtual machine 314 may be a virtual computer system incorporating a computer operating system.

[0055] At step 606, a request to provision an application appliance may be received. For example, the authorized user may submit another provisioning request with the user interface component 404 (Figure 4). In at least one embodiment, the authorized user need not be aware of how the computing application functionality associated with the application appliance is implemented. For example, the authorized user need not be aware that an application appliance instance is provisioned to implement the computing application functionality. The authorized user may request that the computing application functionality be made available to the user VM provisioned at step 602, and the provisioning request of step 606 may be generated in response, for example, as part of an application appliance provisioning workflow. When the application appliance offers one or more optional features, the provisioning request may further specify a set of optional features to activate during provisioning. In at least one embodiment, the provisioning request may further specify a set of optional implementation resources 218 (Figure 2) and/or resource capacities to be made

available to the provisioned application appliance. At step 608, the application appliance may be provisioned. For example, the provisioning component 408 may provision the requested application appliance 320 (Figure 3) in the application vendor space 308 in accordance with the provisioning request of step 606.

5 **[0056]** At step 610, a communication connection between the user VM and the application appliance may be provisioned. For example, the provisioning component 408 (Figure 4) may provision the communication connection 322 (Figure 3) with suitable implementation resources 218 (Figure 2). At step 612, an application appliance access policy set may be configured. For example, the application rights management component 414 may configure
10 the policy enforcement component 410 with one or more policies governing the provisioned application appliance 320, the provisioned user VM 314 and/or the communication connection 322 between them. Alternatively, the application rights management component 414 may provide one or more templates for such policies that are configured by the application appliance provisioning workflow.

15 **[0057]** At step 614, access to the provisioned application appliance in accordance with the access policy set configured at step 612 may be enabled. For example, the policy enforcement component 410 (Figure 4) may begin enforcing the access policy set of step 612, the communication connection 322 (Figure 3) may be activated and/or a local interface corresponding to the interface 332 of the application appliance 320 may be made available to
20 processes maintained by the virtual machine 314.

[0058] Figure 7 depicts example steps for dynamic feature activation in accordance with at least one embodiment. At step 702, a provisioned application appliance instance may subscribe to feature activation requests. For example, the application rights management component 414 may subscribe the application appliance 320 (Figure 3) to such requests. At
25 step 704, a feature activation request may be received. For example, an authorized user associated with a tenant of the virtual resource provider 206 (Figure 2) may request that an optional set of computing application functionality implemented by the application appliance 320 be made available to the user VM 314. The feature activation request may be made through the user interface 404 (Figure 4) and received and processed by the application rights
30 management component 414 and/or the workflow component 416.

[0059] At step 706, the application appliance instance may be notified of the feature activation request received at step 704. For example, the application rights management component 414 (Figure 4) may notify the application appliance 320 (Figure 3) of the feature activation request through a suitable interface element of the application appliance 320. At
5 step 708, a response to the notification of step 706 may be received. For example, the application instance 320 may respond that the requested feature has been activated and/or is available, or else that there was a problem processing the feature activation request.

[0060] At step 710, it may be determined whether the requested feature was activated, for example, in accordance with the response received at step 708. If the requested feature was
10 activated, a process incorporating step 710 may progress to step 714. Otherwise, the process may progress to step 712. At step 712, the sender of the request received at step 704 may be notified of the problem that occurred during processing of the feature activation request. At step 714, the cost tracking component 412 (Figure 4) may be notified of the successful activation of the requested feature, for example, by the application rights management
15 component 414. At step 716, an account associated with the application appliance instance may be updated. For example, the cost tracking component 412 may update a tenant account associated with the user VM 314 to begin accounting for the activated feature in accordance with a corresponding cost plan.

[0061] As described above with reference to Figure 4, the control plane 402 may be
20 facilitated by one or more workflows maintained by the workflow component 416. Figure 8 depicts example steps for workflow management in accordance with at least one embodiment. At step 802, a request may be received by an interface of the control plane 402 (Figure 4). For example, the user interface 404 or the application vendor interface 406 of the control plane 402 may receive the request from a user and/or administrator of the virtual
25 resource provider 202. At step 804, the request may be analyzed to determine one or more actions required to successfully process the request. For example, the provisioning component 408 may analyze the request, and determine a set of actions required to provision a set of computing resources 212 (Figure 2). When an interface element receiving the request corresponds to a specific action to be performed, the interface may extract

information from the request to be utilized in determining aspects and/or parameters of the action to be performed.

[0062] At step 806, a request may be sent to create a workflow based at least in part on the one or more actions determined at step 804. For example, provisioning component 408 (Figure 4) may send the request to the workflow component 416. The request to create the workflow may include the action(s), action metadata such as type of action, and/or action parameters. In at least one embodiment, the control plane 402 and/or the workflow component 416 maintains a job queue for such requests, and workflows are created responsive to new additions to the job queue. At step 808, a workflow and one or more component tasks may be created. For example, the workflow component 416 may analyze the request of step 806 to determine the appropriate workflow and component tasks to create.

[0063] At step 810, execution of the component task(s) may be guided in accordance with the workflow. For example, the workflow component 416 (Figure 4) may activate elements of interfaces of various implementation resources to provision the set of virtual resources. Alternatively, or in addition, the workflow component 416 may manage bids for execution of the component task(s) by components of the virtual resource provider 206 (Figure 2). At step 812, it may be determined whether the workflow has finished. For example, the workflow component 416 may determine whether a final task in a sequence of tasks managed by the workflow has completed. If so, a procedure incorporating step 812 may progress to step 814. Otherwise the procedure may return to step 810 for a next task and/or task sequence. Workflows may guide multiple task sequences executing in parallel. In this case, it may be that the workflow is not finished until each of the multiple task sequences completes and/or an explicit workflow finished flag is set by one of the component tasks. At step 814, the sender of the request of step 802 may be informed of result(s) of the action(s).

[0064] Various embodiments of the disclosure can be described by the following clauses:

Clause 1. A computer-implemented method for managing rights to computing application functionality, comprising:

under control of one or more computer systems configured with executable instructions,

provisioning a first virtual machine that includes an operating system to which at least one user associated with a tenant of a multi-tenant virtual resource provider has access, the provisioning of the first virtual machine facilitated at least in part by a control plane of the multi-tenant virtual resource provider;

5 provisioning a second virtual machine configured at least to execute at least a portion of an application, the provisioning of the second virtual machine facilitated at least in part by the control plane of the multi-tenant virtual resource provider;

providing said at least one user access to functionality of the application at least in part by establishing at least one communication connection between the first virtual machine and the second virtual machine and maintaining at least one interface to the application at the
10 second virtual machine;

enforcing a condition of access to the functionality of the application by said at least one user, the condition of access specifying at least that the access occur through said at least one communication connection and said at least one interface, the enforcing of the condition
15 of access performed at least in part by the control plane of the multi-tenant virtual resource provider; and

permitting data to be conveyed through said at least one communication connection for presentation to said at least one user.

20 Clause 2. A computer-implemented method according to Clause 1, wherein provisioning the first virtual machine and the second virtual machine comprises allocating implementation resources from a pool of implementation resources managed by the control plane of the multi-tenant virtual resource provider.

25 Clause 3. A computer-implemented method according to Clause 2, further comprising: receiving a specification of at least one resource capacity to be available to the application; and

provisioning a set of virtual resources including the second virtual machine with a set of implementation resources from the pool of implementation resources that collectively have a set of resource capacities that include said at least one specified resource capacity.

5 Clause 4. A computer-implemented method according to Clause 3, wherein said at least one specified resource capacity is specified as a multiple of a pre-defined set of implementation resources.

10 Clause 5. A computer-implemented method for managing rights to computing application functionality, comprising:

under control of one or more computer systems configured with executable instructions,

provisioning at least one virtual machine configured at least to execute at least a portion of an application, the provisioning performed at least in part by a virtual resource provider;

15 providing at least one user access to functionality of the application at least in part by establishing at least one communication connection to at least one interface of the application, said at least one interface maintained at least in part by said at least one virtual machine;

20 enforcing a condition of access to the functionality of the application, the condition of access to the functionality of the application specifying at least that the access occur through said at least one communication connection and said at least one interface, the enforcing of the condition of access to the functionality of the application performed at least in part by the virtual resource provider; and

25 permitting data to be conveyed through said at least one communication connection for presentation to said at least one user.

Clause 6. A computer-implemented method according to Clause 5, further comprising enforcing a condition of access to said at least one virtual machine, the condition of access to

said at least one virtual machine specifying at least that the access occur through said at least one communication connection and said at least one interface, the enforcing of the condition of access to said at least one virtual machine performed at least in part by a control plane of the virtual resource provider.

5

Clause 7. A computer-implemented method according to Clause 5, wherein said at least one virtual machine is implemented with a set of implementation resources and access to functionality of the set of implementation resources is controlled by a control plane of the virtual resource provider.

10

Clause 8. A computer-implemented method according to Clause 7, wherein the set of implementation resources includes at least one of: a volatile storage device, a non-volatile storage device, a processor, a physical server, a network interface port, a network switch, and a network path.

15

Clause 9. A computer-implemented method according to Clause 5, wherein said at least one communication connection is implemented with a set of implementation resources and access to functionality of the set of implementation resources is controlled by a control plane of the virtual resource provider.

20

Clause 10. A computer-implemented method according to Clause 5, wherein providing said at least one user access to functionality of the application comprises creating at least one policy specifying the condition of access and enforcing the condition of access comprises enforcing said at least one policy with a policy enforcement component of the virtual resource provider.

25

Clause 11. A computer-implemented method according to Clause 5, wherein said at least one interface comprises a plurality of interface elements corresponding to a plurality of functional features of the application and the condition of access to the functionality of the

application further specifies that the access corresponds to a selected subset of the plurality of interface elements.

5 Clause 12. A computer-implemented method according to Clause 5, wherein provisioning said at least one virtual machine has an associated set of costs that are charged to an account associated with said at least one user.

10 Clause 13. A computer-implemented method according to Clause 12, wherein the set of application costs includes at least one cost corresponding to at least one application feature that is capable of being activated and deactivated.

Clause 14. A computer-implemented method according to Clause 5, further comprising:

receiving, at a control plane of the virtual resource provider, a user request to activate at least one feature of the application from said at least one user;

15 submitting a control plane request to activate said at least one feature to a feature configuration interface of the application, the feature configuration interface maintained at least in part by said at least one virtual machine and inaccessible to said at least one user through said at least one communication connection;

20 receiving, at the control plane, confirmation that said at least one feature has been activated; and

notifying a cost tracking component of the control plane that costs associated with said at least one activated feature are to be charged to an account associated with said at least one user.

25 Clause 15. A computer-implemented method according to Clause 5, wherein provisioning said at least one virtual machine has an associated set of implementation resource costs that are charged to an account associated with said at least one user.

Clause 16. A computer-implemented method according to Clause 5, further comprising receiving a user request to provision said at least one virtual machine, the user request specifying, at least in part, at least one capacity of at least one implementation resource to be made available to said at least one virtual machine.

5

Clause 17. A computer-implemented method for managing rights to computing application functionality, comprising:

under control of one or more computer systems configured with executable instructions,

10 providing at least one user access to functionality of an application at least in part by establishing at least one communication connection to at least one interface of the application, said at least one interface maintained at least in part by at least one virtual machine provisioned at a virtual resource provider;

15 enforcing a condition of access to the functionality of the application, the condition of access to the functionality of the application specifying at least that the access occur through said at least one communication connection and said at least one interface, the enforcing of the condition of access to the functionality of the application performed at least in part by a control plane of the virtual resource provider;

20 tracking at least one cost associated with accessing the functionality of the application through said at least one communication connection and said at least one interface; and

providing tracked cost data for presentation to a tenant of the virtual resource provider.

25 Clause 18. A computer-implemented method according to Clause 17, wherein the functionality of the application is implemented at least in part by at least one implementation resource of the virtual resource provider and tracking said at least one cost comprises tracking a number of time units during which said at least one implementation resource participates in implementing the functionality.

Clause 19. A computer-implemented method according to Clause 17, wherein tracking said at least one cost comprises tracking a number of utilizations of at least one interface element of said at least one interface of the application.

5

Clause 20. A computerized system for managing rights to computing application functionality, comprising:

a set of implementation resources configurable at least to implement a plurality of virtual resources;

10 a virtual resource provisioning component configured at least to provision virtual resources with the set of implementation resources responsive to provisioning requests, the virtual resources including at least one virtual machine configured at least to execute at least a portion of an application and at least one communication connection to at least one interface of the application; and

15 a policy enforcement component configured at least to enforce a condition of access to functionality of the application, the condition of access to the functionality of the application specifying at least that the access occur through said at least one communication connection and said at least one interface.

20 Clause 21. A computerized system according to Clause 20, wherein the computerized system further comprises a user interface component configured at least to enable a user to submit a request to access the functionality of the application and said at least one virtual machine is provisioned by the virtual resource provisioning component at least partly in response to the request to access the functionality of the application.

25

Clause 22. A computerized system according to Clause 20, wherein the computerized system further comprises a vendor interface component configured at least to enable an application vendor to configure said at least one virtual machine to execute said at least a

portion of the application and to configure at least one cost associated with accessing the functionality of the application.

5 Clause 23. One or more computer-readable media having collectively thereon computer-executable instructions that configure one or more computers to collectively, at least:

provision at least one virtual machine configured at least to execute at least a portion of an application, the provisioning facilitated at least in part by a control plane of a virtual resource provider;

10 provide at least one user access to functionality of the application at least in part by establishing at least one communication connection to at least one interface of the application, said at least one interface maintained at least in part by said at least one virtual machine;

enforce a condition of access to the functionality of the application, the condition of access to the functionality of the application specifying at least that the access occur through
15 said at least one communication connection and said at least one interface, the enforcing of the condition of access to the functionality of the application performed at least in part by the control plane of the virtual resource provider; and

permit data to be conveyed through said at least one communication connection for presentation to said at least one user.

20 Clause 24. One or more computer-readable media according to Clause 23, wherein said at least one interface is maintained at a communication network location that is remote with respect to an operating system of said at least one user and access to said at least one communication connection is through at least one corresponding interface that is local to the
25 operating system.

Clause 25. One or more computer-readable media according to Clause 23, wherein said at least one interface comprises an interface in accordance with a remote desktop protocol.

The various embodiments described herein may be implemented in a wide variety of operating environments, which in some cases may include one or more user computers, computing devices, or processing devices which may be utilized to operate any of a number of applications. User or client devices may include any of a number of general purpose
5 personal computers, such as desktop or laptop computers running a standard operating system, as well as cellular, wireless, and handheld devices running mobile software and capable of supporting a number of networking and messaging protocols. Such a system also may include a number of workstations running any of a variety of commercially-available operating systems and other known applications for purposes such as development and
10 database management. These devices also may include other electronic devices, such as dummy terminals, thin-clients, gaming systems, and other devices capable of communicating via a network.

[0065] Most embodiments utilize at least one network that would be familiar to those skilled in the art for supporting communications using any of a variety of commercially-
15 available protocols, such as TCP/IP, OSI, FTP, UPnP, NFS, CIFS, and AppleTalk. Such a network may include, for example, a local area network, a wide-area network, a virtual private network, the Internet, an intranet, an extranet, a public switched telephone network, an infrared network, a wireless network, and any combination thereof. The network may, furthermore, incorporate any suitable network topology. Examples of suitable network
20 topologies include, but are not limited to, simple point-to-point, star topology, self organizing peer-to-peer topologies, and combinations thereof.

[0066] In embodiments utilizing a Web server, the Web server may run any of a variety of server or mid-tier applications, including HTTP servers, FTP servers, CGI servers, data servers, Java servers, and business application servers. The server(s) also may be capable of
25 executing programs or scripts in response requests from user devices, such as by executing one or more Web applications that may be implemented as one or more scripts or programs written in any programming language, such as Java[®], C, C# or C++, or any scripting language, such as Perl, Python, or TCL, as well as combinations thereof. The server(s) may also include database servers, including without limitation those commercially available
30 from Oracle[®], Microsoft[®], Sybase[®], and IBM[®].

[0067] The environment may include a variety of data stores and other memory and storage media as discussed above. These may reside in a variety of locations, such as on a storage medium local to (and/or resident in) one or more of the computers or remote from any or all of the computers across the network. In a particular set of embodiments, the information may reside in a storage-area network (“SAN”) familiar to those skilled in the art. Similarly, any necessary files for performing the functions attributed to the computers, servers, or other network devices may be stored locally and/or remotely, as appropriate. Where a system includes computerized devices, each such device may include hardware elements that may be electrically coupled via a bus, the elements including, for example, at least one central processing unit (CPU), at least one input device (e.g., a mouse, keyboard, controller, touch screen, or keypad), and at least one output device (e.g., a display device, printer, or speaker). Such a system may also include one or more storage devices, such as disk drives, optical storage devices, and solid-state storage devices such as random access memory (“RAM”) or read-only memory (“ROM”), as well as removable media devices, memory cards, flash cards, etc.

[0068] Such devices also may include a computer-readable storage media reader, a communications device (e.g., a modem, a network card (wireless or wired), an infrared communication device, etc.), and working memory as described above. The computer-readable storage media reader may be connected with, or configured to receive, a computer-readable storage medium, representing remote, local, fixed, and/or removable storage devices as well as storage media for temporarily and/or more permanently containing, storing, transmitting, and retrieving computer-readable information. The system and various devices also typically will include a number of software applications, modules including program modules, services, or other elements located within at least one working memory device, including an operating system and application programs, such as a client application or Web browser. It should be appreciated that alternate embodiments may have numerous variations from that described above. For example, customized hardware might also be utilized and/or particular elements might be implemented in hardware, software (including portable software, such as applets), or both. Further, connection to other computing devices such as network input/output devices may be employed.

[0069] Storage media and computer readable media for containing code, or portions of code, may include any appropriate media known or used in the art, including storage media and communication media, such as but not limited to volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage and/or transmission of information such as computer readable instructions, data structures, program modules, or other data, including RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disk (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which may be utilized to store the desired information and which may be accessed by the a system device. Program modules, program components and/or programmatic objects may include computer-readable and/or computer-executable instructions of and/or corresponding to any suitable computer programming language. In at least one embodiment, each computer-readable medium may be tangible. In at least one embodiment, each computer-readable medium may be non-transitory in time. Based on the disclosure and teachings provided herein, a person of ordinary skill in the art will appreciate other ways and/or methods to implement the various embodiments.

[0070] The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will, however, be evident that various modifications and changes may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the claims.

[0071] The use of the terms “a” and “an” and “the” and similar referents in the context of describing embodiments (especially in the context of the following claims) are to be construed to cover both the singular and the plural, unless otherwise indicated herein or clearly contradicted by context. The terms “comprising,” “having,” “including,” and “containing” are to be construed as open-ended terms (i.e., meaning “including, but not limited to,”) unless otherwise noted. The term “connected” is to be construed as partly or wholly contained within, attached to, or joined together, even if there is something intervening. Recitation of ranges of values herein are merely intended to serve as a shorthand method of referring individually to each separate value falling within the range, unless otherwise indicated herein, and each separate value is incorporated into the

specification as if it were individually recited herein. All methods described herein can be performed in any suitable order unless otherwise indicated herein or otherwise clearly contradicted by context. The use of any and all examples, or exemplary language (*e.g.*, “such as”) provided herein, is intended merely to better illuminate embodiments and does not
5 pose a limitation on the scope unless otherwise claimed. No language in the specification should be construed as indicating any non-claimed element as essential to the practice of at least one embodiment.

[0072] Preferred embodiments are described herein, including the best mode known to the inventors. Variations of those preferred embodiments may become apparent to those of
10 ordinary skill in the art upon reading the foregoing description. The inventors expect skilled artisans to employ such variations as appropriate, and the inventors intend for embodiments to be constructed otherwise than as specifically described herein. Accordingly, suitable embodiments include all modifications and equivalents of the subject matter recited in the claims appended hereto as permitted by applicable law. Moreover, any combination of the
15 above-described elements in all possible variations thereof is contemplated as being incorporated into some suitable embodiment unless otherwise indicated herein or otherwise clearly contradicted by context.

[0073] All references, including publications, patent applications, and patents, cited herein are hereby incorporated by reference to the same extent as if each reference were
20 individually and specifically indicated to be incorporated by reference and were set forth in its entirety herein.

WHAT IS CLAIMED IS:

1 1. A computer-implemented method for managing rights to computing
2 application functionality, comprising:
3 under control of one or more computer systems configured with executable
4 instructions,
5 provisioning at least one virtual machine configured at least to execute at least
6 a portion of an application, the provisioning performed at least in part by a virtual resource
7 provider;
8 providing at least one user access to functionality of the application at least in
9 part by establishing at least one communication connection to at least one interface of the
10 application, said at least one interface maintained at least in part by said at least one virtual
11 machine;
12 enforcing a condition of access to the functionality of the application, the
13 condition of access to the functionality of the application specifying at least that the access
14 occur through said at least one communication connection and said at least one interface, the
15 enforcing of the condition of access to the functionality of the application performed at least
16 in part by the virtual resource provider; and
17 permitting data to be conveyed through said at least one communication
18 connection for presentation to said at least one user.

1 2. A computer-implemented method according to Claim 1, further
2 comprising enforcing a condition of access to said at least one virtual machine, the condition
3 of access to said at least one virtual machine specifying at least that the access occur through
4 said at least one communication connection and said at least one interface, the enforcing of
5 the condition of access to said at least one virtual machine performed at least in part by a
6 control plane of the virtual resource provider.

1 3. A computer-implemented method according to Claim 1, wherein said at
2 least one virtual machine is implemented with a set of implementation resources and access
3 to functionality of the set of implementation resources is controlled by a control plane of the
4 virtual resource provider.

1 4. A computer-implemented method according to Claim 3, wherein the set
2 of implementation resources includes at least one of: a volatile storage device, a non-volatile
3 storage device, a processor, a physical server, a network interface port, a network switch, and
4 a network path.

1 5. A computer-implemented method according to Claim 1, wherein said at
2 least one communication connection is implemented with a set of implementation resources
3 and access to functionality of the set of implementation resources is controlled by a control
4 plane of the virtual resource provider.

1 6. A computer-implemented method according to Claim 1, wherein
2 providing said at least one user access to functionality of the application comprises creating
3 at least one policy specifying the condition of access and enforcing the condition of access
4 comprises enforcing said at least one policy with a policy enforcement component of the
5 virtual resource provider.

1 7. A computer-implemented method according to Claim 1, wherein said at
2 least one interface comprises a plurality of interface elements corresponding to a plurality of
3 functional features of the application and the condition of access to the functionality of the
4 application further specifies that the access corresponds to a selected subset of the plurality
5 of interface elements.

1 8. A computer-implemented method according to Claim 1, wherein
2 provisioning said at least one virtual machine has an associated set of costs that are charged
3 to an account associated with said at least one user.

1 9. A computer-implemented method according to Claim 8, wherein the set
2 of application costs includes at least one cost corresponding to at least one application feature
3 that is capable of being activated and deactivated.

1 10. A computer-implemented method according to Claim 1, further
2 comprising:

3 receiving, at a control plane of the virtual resource provider, a user request to
4 activate at least one feature of the application from said at least one user;

5 submitting a control plane request to activate said at least one feature to a
6 feature configuration interface of the application, the feature configuration interface
7 maintained at least in part by said at least one virtual machine and inaccessible to said at least
8 one user through said at least one communication connection;

9 receiving, at the control plane, confirmation that said at least one feature has
10 been activated; and

11 notifying a cost tracking component of the control plane that costs associated
12 with said at least one activated feature are to be charged to an account associated with said at
13 least one user.

1 11. A computer-implemented method according to Claim 1, wherein
2 provisioning said at least one virtual machine has an associated set of implementation
3 resource costs that are charged to an account associated with said at least one user.

1 12. A computer-implemented method according to Claim 1, further
2 comprising receiving a user request to provision said at least one virtual machine, the user
3 request specifying, at least in part, at least one capacity of at least one implementation
4 resource to be made available to said at least one virtual machine.

1 13. A computerized system for managing rights to computing application
2 functionality, comprising:

3 a set of implementation resources configurable at least to implement a
4 plurality of virtual resources;

5 a virtual resource provisioning component configured at least to provision
6 virtual resources with the set of implementation resources responsive to provisioning
7 requests, the virtual resources including at least one virtual machine configured at least to
8 execute at least a portion of an application and at least one communication connection to at
9 least one interface of the application; and

10 a policy enforcement component configured at least to enforce a condition of
11 access to functionality of the application, the condition of access to the functionality of the

12 application specifying at least that the access occur through said at least one communication
13 connection and said at least one interface.

1 14. A computerized system according to Claim 13, wherein the
2 computerized system further comprises a user interface component configured at least to
3 enable a user to submit a request to access the functionality of the application and said at
4 least one virtual machine is provisioned by the virtual resource provisioning component at
5 least partly in response to the request to access the functionality of the application.

1 15. A computerized system according to Claim 13, wherein the
2 computerized system further comprises a vendor interface component configured at least to
3 enable an application vendor to configure said at least one virtual machine to execute said at
4 least a portion of the application and to configure at least one cost associated with accessing
5 the functionality of the application.

1

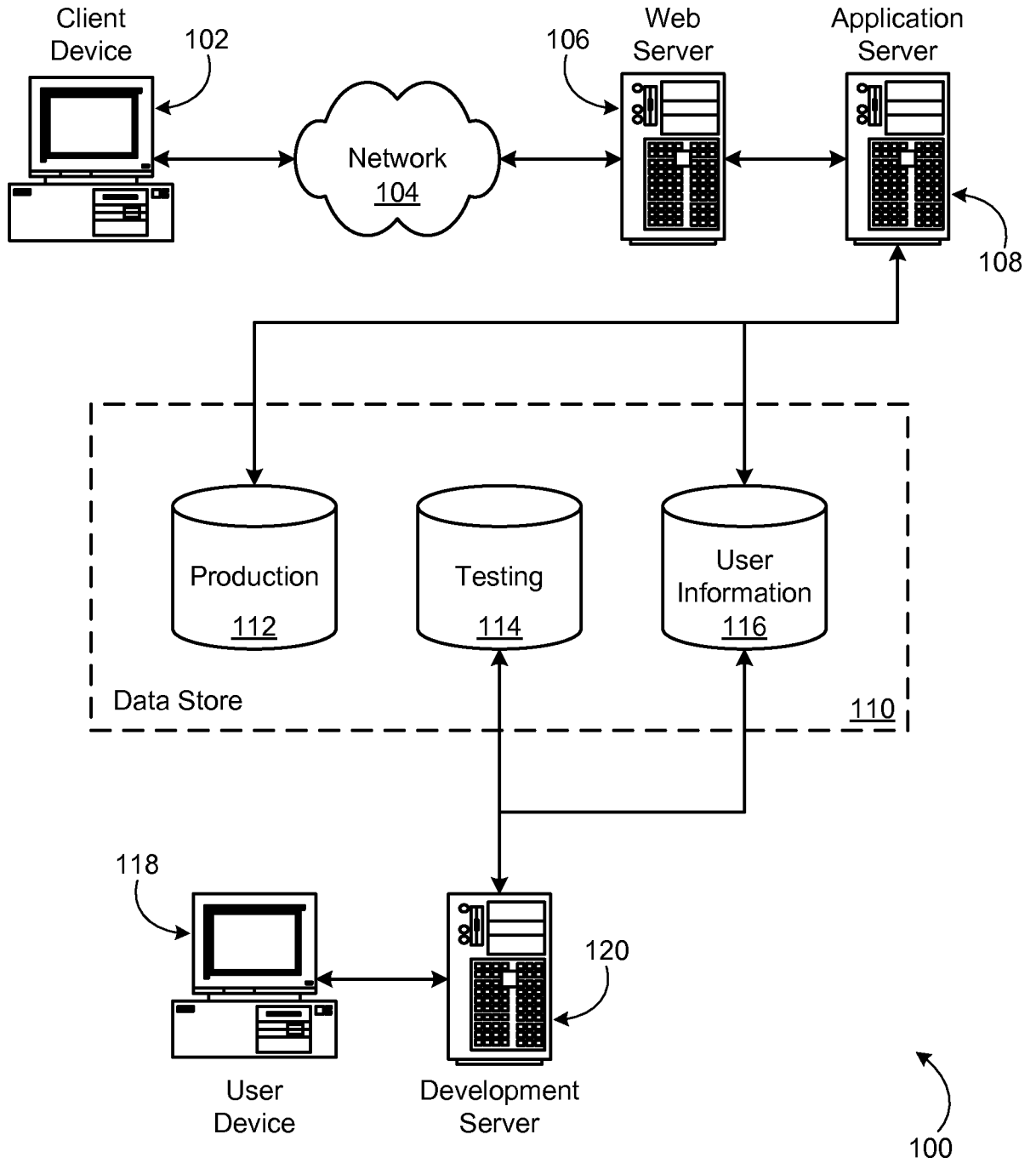


Figure 1

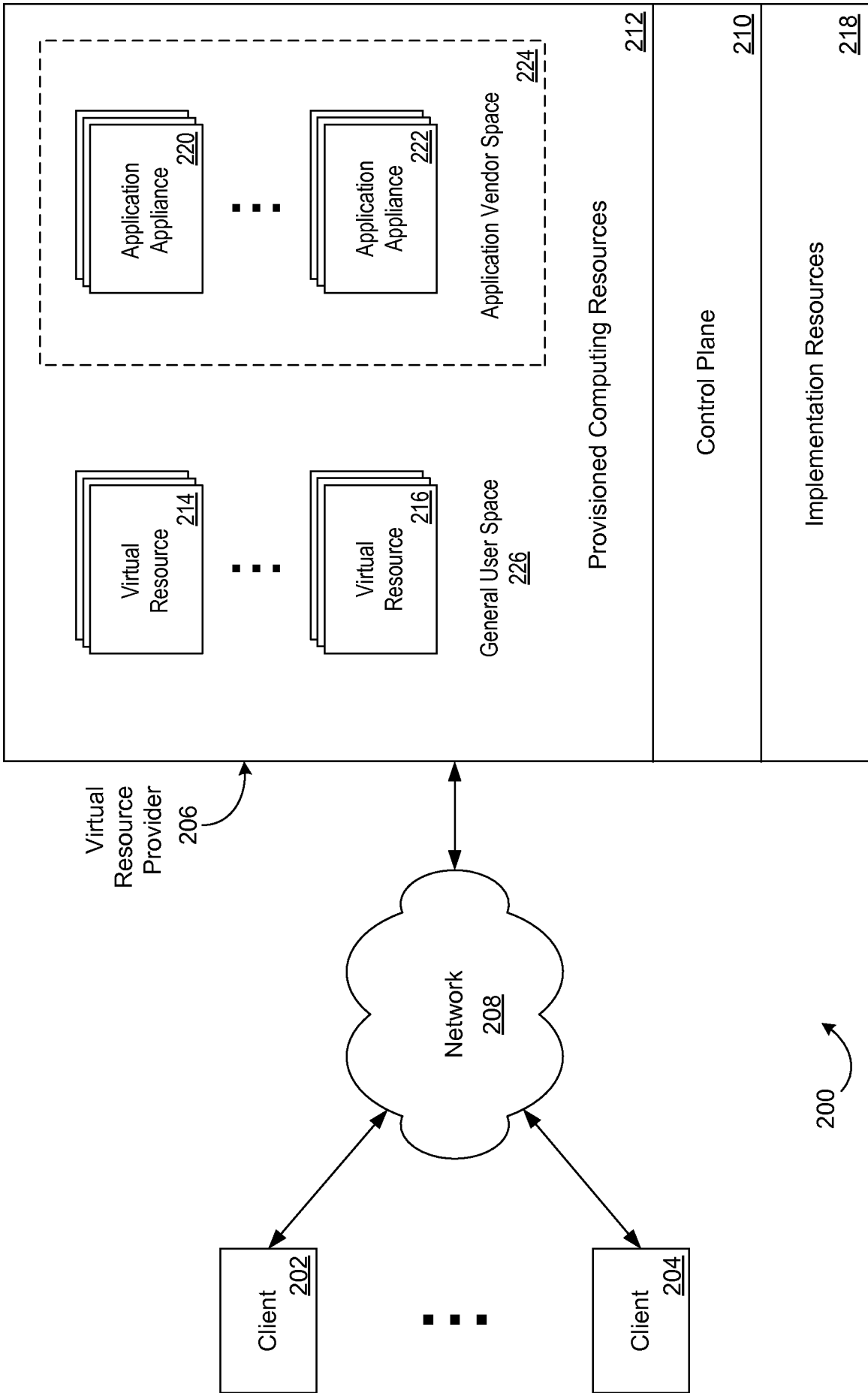


Figure 2

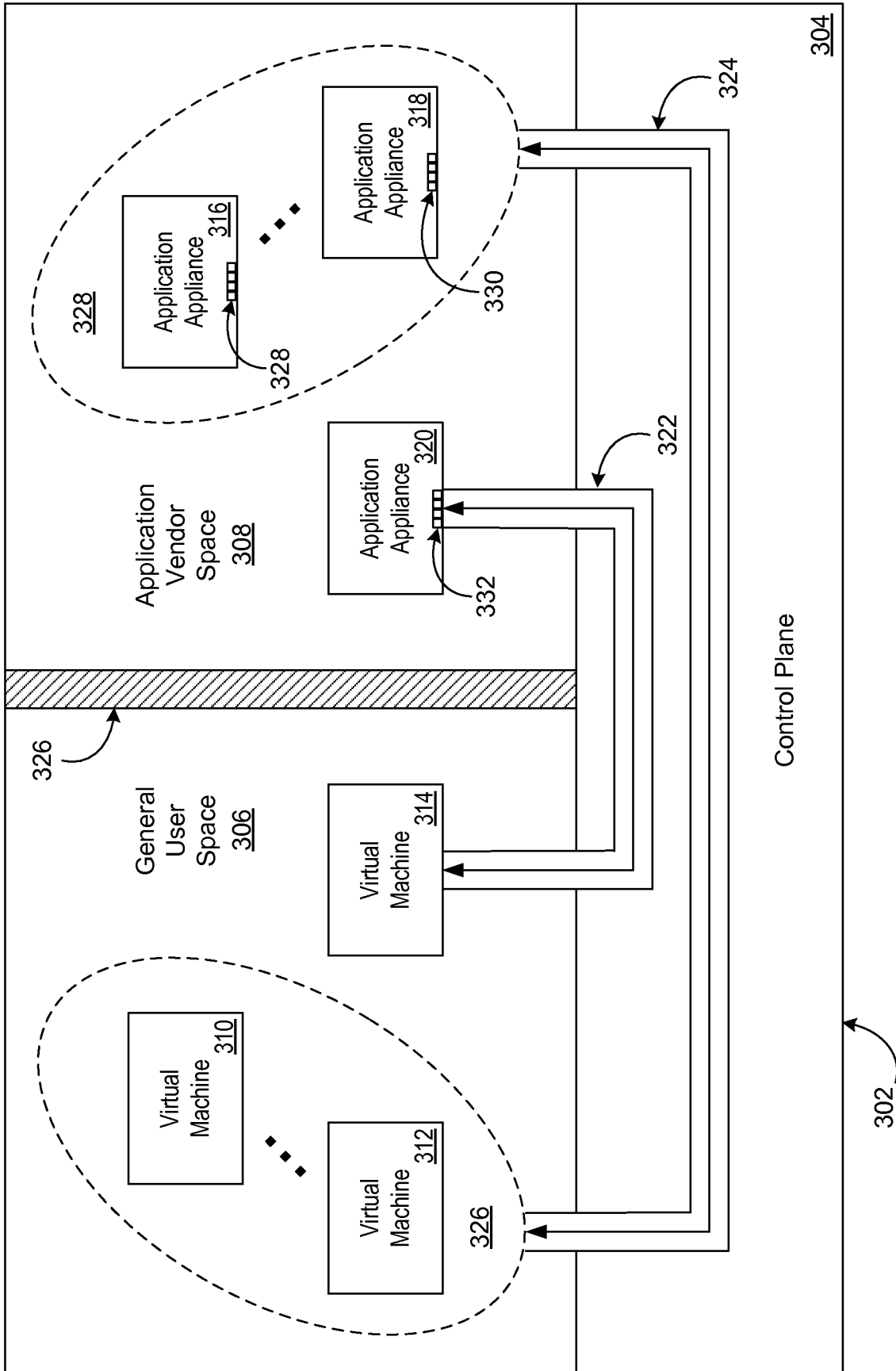


Figure 3

