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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 March 2012 (08.03.2012)

PCT

(10) International Publication Number
WO 2012/030624 A1

(51) International Patent Classification:
G06F 21/00 (2006.01)

(21) International Application Number:
PCT/US2011/049174

(22) International Filing Date:
25 August 2011 (25.08.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/378,294 30 August 2010 (30.08.2010) US

(71) Applicant *(for all designated States except US)*: **VMWARE, INC.** [US/US]; 3401 Hillview Avenue, Palo Alto, CA 94304 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **PUGH, William** [US/US]; 615 2nd Avenue, Suite 300, Seattle, WA 98104 (US). **EAKES, Michael** [US/US]; 3401 Hillview Avenue, Palo Alto, CA 94304 (US). **BERNANDER, Ojvind** [US/US]; 3401 Hillview Avenue, Palo Alto, CA 94304 (US). **GUNDAVARAM, Pradyumna** [IN/US]; 3401 Hillview Avenue, Palo Alto, CA 94304 (US).

(74) Agents: **HEYMAN, Leonard, E.** et al.; VMware, Inc., 3401 Hillview Avenue, Palo Alto, CA 94304 (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, 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, 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,

[Continued on next page]

(54) Title: UNIFIED WORKSPACE FOR THIN, REMOTE, AND SAAS APPLICATIONS

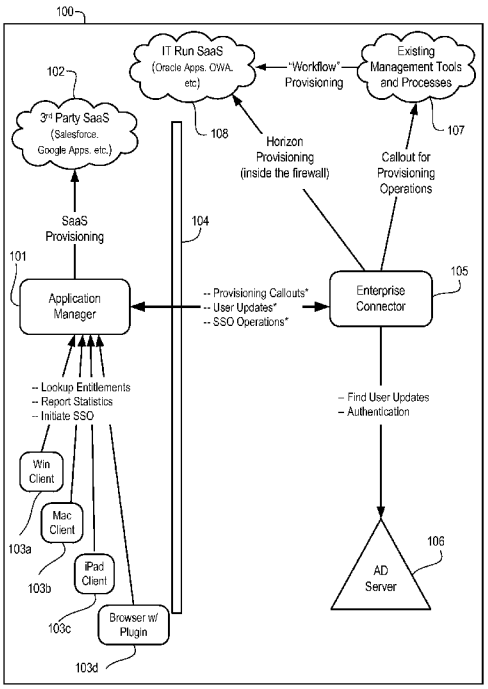

Fig. 1

(57) Abstract: Application-manager software authenticates a user of a client device over a channel. The authentication operation is performed using a directory service. The application-manager software presents a plurality of applications in a GUI displayed by the client device. The plurality of applications depends on the authentication, the client device, and the channel. And the plurality of applications includes a thin application and a software-as-a-service (SaaS) application. The application-manager software receives a selection as to an application from the user. If the selection is for the SaaS application, the application-manager software provisions the SaaS application. The provision includes automatically logging the user onto an account with a provider of the SaaS application using a single sign-on and connecting the user to the account so that the user can interact with the SaaS application. If the selection is for the thin application, the application manager software launches the thin application.

WO 2012/030624 A1

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).

Published:

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

UNIFIED WORKSPACE FOR THIN, REMOTE, AND SAAS APPLICATIONS

RELATED APPLICATION

[0001] This application is related to U.S. Patent 7,779,091, entitled "Method And System For Providing Virtualized Application Workspaces", which issued on August 17, 2010. The disclosure of that application is also hereby incorporated by reference.

BACKGROUND

[0002] In the context of deploying software applications at the enterprise level, the term "entitlement" refers to the process of determining if a user is permitted access to an application, e.g., according to a licensing arrangement or a company policy. The term "provisioning" refers to the process of actually giving a user an application, or making it accessible to the user. Provisioning generally occurs only after entitlement has been determined.

[0003] Traditionally, desktop-application entitlement and provisioning was accomplished by running an installer program or script to install an application on a computing system (e.g., a hardware device with platform software) associated with a user who has a license/permission to use the application. Often, such an installation can be automated.

[0004] Some applications, referred to herein as "thin applications," do not require a full-blown installation (e.g., including registry settings) on the user's computing system and can run from a simple executable file that merely needs to be copied to the user's disk. For example, thin applications include many virtualized applications, such as those virtualized using ThinApp™, available from VMware, Inc. of Palo Alto, California. Similarly, other applications may be accessed remotely over a network, and need very little, if any, customization of the user's device. For example, applications may be accessed remotely using a virtual desktop infrastructure (VDI) solution, such as VMware View™, a desktop-remoting system that includes a thin client accessed directly by a user, which remotes to a desktop from a remote virtual machine. In another example, a terminal server hosting user applications may be accessed using a browser or other thin client.

[0005] Likewise, software as a Service (SaaS) applications, by their nature as cloud-based applications, generally do not require installation or modification of the user's system or device. SaaS applications are generally web-based, e.g., accessed and used through a web browser. This simplifies provisioning, but can complicate both authentication and entitlement enforcement. For a SaaS application, a user typically authenticates with the SaaS provider, e.g., by providing a user identifier and password combination, before the SaaS will connect the user to his/her account and grant access to a particular application.

[0006] As software development shifts away from traditional applications to virtualized applications, remote applications, and SaaS applications, a need arises for a centralized approach to managing the entitlement and provisioning of such applications on a growing number of client devices that use a growing number of access channels (e.g., mobile devices such as smartphones that use Wi-Fi networks).

SUMMARY

[0007] In an example embodiment, a processor-executed method is described for providing applications to a user. According to the method, application-manager software authenticates a user of a client device over a channel. The authentication operation is performed using a directory service. The application-manager software presents a plurality of applications in a graphical user interface (GUI) displayed by the client device. The plurality of applications depends at least in part on the authentication, the client device, and the channel. And the plurality of applications includes a thin application and a software-as-a-service (SaaS) application. The application-manager software receives a selection as to an application from the user. If the selection is for the SaaS application, the application-manager software provisions the SaaS application. The provision includes automatically logging the user onto an account with a provider of the SaaS application using a single sign-on and connecting the user to the account so that the user can interact with the SaaS application. If the selection is for the thin application, the application manager software launches the thin application.

[0008] In another example embodiment, an apparatus is described, namely, a computer-readable storage medium that persistently stores a program for providing applications to a user. The program authenticates a user of a client device over a channel. The authentication operation is performed using a directory service. The program presents a plurality of applications in a GUI displayed by the client device. The plurality of applications

depends at least in part on the authentication, the client device, and the channel. And the plurality of applications includes a thin application and a software-as-a-service (SaaS) application. The program receives a selection as to an application from the user. If the selection is for the SaaS application, the program provisions the SaaS application. The provision includes automatically logging the user onto an account with a provider of the SaaS application using a single sign-on and connecting the user to the account so that the user can interact with the SaaS application. If the selection is for the thin application, the program launches the thin application.

[0009] Another example embodiment also involves a processor-executed method for providing applications to a user. According to the method, application-manager software authenticates a user of a client device over a channel. The authentication operation is performed using a directory service. The application-manager software presents a plurality of applications in a GUI displayed by the client device. The plurality of applications depends at least in part on the authentication, the client device, and a geolocation of the client device. And the plurality of applications includes a sandboxed application and a software-as-a-service (SaaS) application. The application-manager software receives a selection as to an application from the user. If the selection is for the SaaS application, the application-manager software provisions the SaaS application. The provision includes automatically logging the user onto an account with a provider of the SaaS application using a single sign-on and connecting the user to the account so that the user can interact with the SaaS application. If the selection is for the sandboxed application, the application manager software launches the sandboxed application.

[0010] Other aspects and advantages of the inventions will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, which illustrate by way of example the principles of the inventions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Figure 1 is a simplified system diagram that illustrates an architecture for a system for managing authentication, entitlements, and provisioning, in accordance with an example embodiment.

[0012] Figure 2 shows a diagram of the functional modules in a system for managing authentication, entitlements, and provisioning, in accordance with an example embodiment.

[0013] Figure 3A is a sequence diagram depicting a scenario for SSO, in accordance with an example embodiment.

[0014] Figure 3B is a sequence diagram depicting another scenario for SSO, in accordance with an example embodiment.

[0015] Figure 3C is a sequence diagram depicting another scenario for SSO, in accordance with an example embodiment.

[0016] Figure 4 is a sequence diagram showing a process for entitling and provisioning a new SaaS account, in accordance with an example embodiment.

[0017] Figure 5 is a flowchart diagram that illustrates a process for storing provisioning tasks, in accordance with an example embodiment.

[0018] Figure 6 is a flowchart diagram that illustrates a process for determining provisioning tasks, in accordance with an example embodiment.

[0019] Figure 7 is a flowchart diagram that illustrates a process for provisioning an application, in accordance with an example embodiment.

[0020] Figure 8 is a flowchart diagram that illustrates a process for provisioning the Mozy application, in accordance with an example embodiment.

[0021] Figure 9A and 9B are flowchart diagrams that illustrate a process for authenticating a user to a SaaS application from a session, in accordance with an example embodiment.

[0022] Figure 10 shows a GUI view of a dashboard tab displayed by an administrator web portal, in accordance with an example embodiment.

[0023] Figure 11 shows a GUI view of a users tab displayed by an administrator web portal, in accordance with an example embodiment.

[0024] Figure 12 shows a GUI view of a user edit dialog displayed by an administrator web portal, in accordance with an example embodiment.

[0025] Figure 13 shows a GUI view of a groups tab displayed by an administrator web portal, in accordance with an example embodiment.

[0026] Figure 14 shows a GUI view of a group edit dialog displayed by an administrator web portal, in accordance with an example embodiment.

[0027] Figure 15 shows a GUI view of an entitlement edit dialog displayed by an administrator web portal, in accordance with an example embodiment.

[0028] Figure 16 shows a GUI view of an applications tab displayed by an administrator web portal, in accordance with an example embodiment.

[0029] Figure 17 shows a GUI view of a dialog for adding an application, in accordance with an example embodiment.

[0030] Figure 18 shows a GUI view of an edit dialog for editing a Windows desktop application, in accordance with an example embodiment.

[0031] Figure 19 shows a GUI view of an edit dialog for editing a published (or remote) application, in accordance with an example embodiment.

[0032] Figure 20 shows a GUI view of a licenses tab displayed by an administrator web portal, in accordance with an example embodiment.

[0033] Figure 21 shows a GUI view of a policy tab displayed by an administrator web portal, in accordance with an example embodiment.

[0034] Figure 22 shows a GUI view displayed by a user web portal, in accordance with an example embodiment.

[0035] Figure 23 shows another GUI view displayed by a user web portal, in accordance with an example embodiment.

DETAILED DESCRIPTION

[0036] In the following description, numerous specific details are set forth in order to provide a thorough understanding of the exemplary embodiments. However, it should be noted that the example embodiments described herein may be practiced without some of these specific details. In other instances, process operations and implementation details have not been described in detail where they are already well known.

[0037] Generally speaking, the example embodiments described herein provide a unified workspace for desktop (both local and remote) applications, software-as-a-service (SaaS) applications, and data. End users are provided with a single workspace that works across various client devices to access applications, while providing administrators a single place for managing application entitlement and provisioning.

[0038] Figure 1 is a simplified system diagram that illustrates an architecture for a system 100 for managing authentication, entitlements, and provisioning, in accordance with an example embodiment. As depicted in this figure, the system includes an Application Manager 101, which, among other things, allows: (1) administrators to connect (e.g., through an administrator portal) to manage users, applications and accounts; (2) end users to connect (e.g., through a user portal), where they can (a) launch applications, in a relatively seamless manner, or (b) enable and disable applications to which they are entitled; and (c) employ client devices to connect through a RESTful API and to get lists of application entitlements per user/device. It will be appreciated that "REST" is an acronym for "Representational State Transfer" and "API" is an acronym for "application programming interface." It will further be appreciated that a RESTful API typically supports HTTP (hypertext transfer protocol) operations (e.g., POST, GET, PUT or DELETE) on an Internet media type (e.g., JSON (JavaScript Object Notation), XML (eXtensible Markup Language), or YAML).

[0039] As depicted in Figure 1, Application Manager 101 communicates (e.g., over a network which may include the Internet or other WAN) with 3rd Party SaaS 102, which are SaaS application providers (such as Salesforce.com, Google, etc.) supported by the system. Application Manager 101 can provision new user accounts with 3rd Party SaaS 102 and provide single sign-on (SSO). In an example embodiment, 3rd Party SaaS 102 might have a provisioning API and/or support SAML (Security Assertion Markup Language). SAML is a well-known standard for enabling SSO authentication. For 3rd Party SaaS 102 providers that

do not support SAML or have a provisioning API, Application Manager 101 might provide down-level screen scraping APIs to enable the provisioning of new user accounts.

[0040] As further depicted in Figure 1, Application Manager 101 communicates (e.g., over the Internet or other WAN which might be wireless in whole or in part) with supported client devices, including, by way of example without limitation: (1) Windows Client 103a (e.g., with a platform including an Intel x86-family microprocessor running a Microsoft Windows® operating system); (2) Mac Client 103b (e.g., with a platform including an Intel x86-family microprocessor running a Mac™ OS operating system); and (3) iPad® Client 103c (e.g., with a platform including an ARM microprocessor running an iOS™ operating system). In an example embodiment, this communication takes place through calls by native agents on the supported client devices to the RESTful API described above. For example, a native agent might periodically use the RESTful API to transmit device configuration information relating to: (a) the client device's operating system (and patches); (b) whether virus protection/firewall/anti-spyware is installed; (c) the domain to which the client device is attached; and (c) whether auto-updates are installed. In response, the native agent would receive a list of entitlements from Application Manager 101. Then the native agent would compare the entitlements to what is currently installed and execute the appropriate install, upgrade, and/or uninstall operations.

[0041] For thin applications and native applications, installing might entail installing an MSI (Windows Installer). For remote applications, installing might entail setting up the RDP (Remote Desktop Protocol). For SaaS applications, installing might entail setting up a URL shortcut to the application. The shortcut may be installed to the user's desktop and appear as an icon that can be activated by double-clicking. The URL (Uniform Resource Locator) for this shortcut will launch the user's browser to access Application Manager 101, with a redirect portion of the URL for the SaaS application. More specifically, when a user clicks on the SaaS shortcut, the URL of the shortcut will direct to Application Manager 101 to initiate an SSO connection. After the SSO session is created, the Application Manager 101 will redirect user's browser to the 3rd Party SaaS 102 with the appropriate authentication information as specified by the SAML IdP-initiated SSO profile.

[0042] As also depicted in Figure 1, Application Manager 101 can also communicate with web browser 103d (e.g., a browser which supports HTML5, Adobe Flash, etc.). In an example embodiment, the system provides a user portal that can be accessed directly by a web browser with a system plug-in, rather than a native agent. The functionality included in

such a system plug-in might, for example: (1) enable SSO for down-level SaaS applications that do not support SAML (or any other standard for an ID protocol); and (2) provide metrics to the system on SaaS application usage and/or geolocation for reporting to administrators. Here it will be appreciated that geolocation information might be obtained from a client device through the use of the W3C Geolocation API or a similar API in a web browser, in an example embodiment.

[0043] Additionally, Figure 1 depicts Application Manager 101 communicating across firewall 104 with Enterprise Connector 105. In an example embodiment, Enterprise Connector 105 might initiate a secure connection to Application Manager 101 so that Application Manager 101 can send commands to Enterprise Connector 105. Also, Enterprise Connector 105 might be a virtual appliance, in an example embodiment. Enterprise Connector 105 provides a bridge into the enterprise for integration with existing corporate processes and directories. To prevent companies from having to open holes in their firewall, Enterprise Connector 105 is installed behind firewall 104 and is connected to the Application Manager 101, e.g., using a secure connection or using a local area network (LAN) or wide area network (WAN) which might be wireless in whole or in part, but which employs security measures to vouch for identity. An example of such a security measure is a certificate installed on Enterprise Connector 105 and registered with Application Manager 101, so that Application Manager 101 can verify the identity of Enterprise Connector 105.

[0044] In an example embodiment, Enterprise Connector 105 might act as an Identity Provider (IdP) for SSO. In this capacity, Enterprise Connector 105 might provide authentication against an internal LDAP (Lightweight Directory Access Protocol) server such as Active Directory ("AD") Server 106, without exposing passwords. It will be appreciated that in such an embodiment, AD Server 106 is the "source of truth" for corporate user identity and as such receives and reports updates to corporate user identity. In an alternative example embodiment where Enterprise Connector 105 is absent, Application Manager 101 might directly connect, using a connection which is secure or which employs security measures to AD Server 106, when performing authentication and operations involving user updates.

[0045] Additionally, Enterprise Connector 105 might act as a staging area behind firewall 104 for integrating with existing workflow and licensing processes. In this capacity, Enterprise Connector 105 might make provisioning callouts to existing management tools and processes 107, using, for example, custom workflows built on WS-BPEL (Web Services

Business Process Execution Language) or built into management tools. When making these provisioning callouts, Enterprise Connector 105 will utilize information from existing license tracking systems, asset management systems, etc.

[0046] Similarly, Enterprise Connector 105 might perform SaaS provisioning inside firewall 104, using SAML or provisioning APIs exposed by IT Run SaaS 108. IT Run SaaS 108 might include SaaS applications such as Oracle Apps, Outlook Web App (OWA), etc. In an example embodiment, some or all of these internal SaaS applications might not be provisioned against AD Server 106.

[0047] Enterprise Connector 105 may allow extensions to system 100 for supporting additional applications. For example, a user of system 100 might want to use Jira®, an issue and project tracking application available from Atlassian Pty Ltd. of Sydney, Australia, but system 100 does not provide drivers to support such use. In this scenario, the user may create their own drivers and install them, e.g., as plugins or extensions, in Enterprise Connector 105. The drivers will then run as part of any provisioning operation involving Jira.

[0048] In an example embodiment, the system depicted in Figure 1 might run on a number of servers connected by a network (e.g., a LAN or a WAN) to each other in a cluster or other distributed system which might execute distributed-computing software (including Apache's Hadoop or other software based on Map-Reduce and/or Google File System), virtualization software (e.g., as provided by VMware, Citrix, Microsoft, etc.), load-balancing software, database software (e.g., SQL, NoSQL, etc.), web server software, etc. In turn, the distributed system might be connected (e.g., by a storage area network (SAN)) to persistent storage which stores (e.g., in a database or other file) data related to authentication, entitlements, and provisioning. The servers themselves might include: (1) hardware consisting of one or more microprocessors (e.g., from the x86 family), volatile storage (e.g., RAM), and persistent storage (e.g., a hard disk or solid-state drive); and (2) an operating system (e.g., Linux, Windows Server, Mac OS Server, etc.) that runs on the hardware.

[0049] Figure 2 shows a diagram of the functional modules in a system 200 for managing authentication, entitlements, and provisioning, in accordance with an example embodiment. As depicted in this figure, two of the functional modules reside on client devices: (1) the browser plug-in (or agent) 103e, which interfaces with a web browser 103d and that reports usage to RESTful API 203 for logging and reporting by the system as described above; and (2) the native agent (or client) 103, also described above, which is

installed on a client device with an x86 platform (e.g., a Windows or Mac laptop or desktop) or an ARM platform (e.g., an iPhone, an iPad, an Android smartphone, an Android tablet, etc.).

[0050] As depicted in Figure 2, the server-side modules of the system include four modules that interface with the client-side modules. One of these modules is the Administrative Web UI (user interface) 201, where administrators can go to manage, among other things: (1) applications and application families (e.g., where a "family" is an arbitrary group of applications that may be managed as a unit, as opposed to a family of different versions of a single application); (2) end users; (3) client devices and policies related to client devices; and (4) entitlements. Also, administrators might be able to view reports on usage and/or geolocation at Administrative Web UI 201, in an example embodiment. Elsewhere in the disclosure, Administrative Web UI 201 is called the "administrator web portal".

[0051] Further, as depicted in Figure 2, the server-side modules of the system include a User Web UI 202 where users can go to launch applications in a relatively seamless manner and enable disabled applications they are entitled to. Also, the system's server-side modules include a RESTful API 203, which may include a set of APIs and provides access to some or all of the system's functionality, depending on how the system is implemented and/or configured. In an alternative example embodiment, the system's functionality might be accessed in whole or in part using the SOAP (Simple Object Access Protocol) protocol or some other protocol for inter-process communication. Elsewhere in the disclosure, User Web UI 201 is called the "user web portal".

[0052] As additionally depicted in Figure 2, the server-side modules of the system include a SAML Front End 204, which understands SAML and which creates the SAML session, using SSO Service 211, to perform most of the actual work related to SSO authentication.

[0053] Beneath the four server-side modules that interface with the client-side modules is an Internal Java API 205 which server-side modules 201-204 use to communicate and to access the server-side modules that provide back-end functionality. The Internal Java API 205 might log all of its API calls to the Logging/Analytics Store 215, in an example embodiment. This information might then be used for generating user reports on usage and/or geolocation.

[0054] The server-side modules that provide back-end functionality include a module 206 for account management which provides an API for purchasers of the system to interact with the entity distributing the system (i.e., VMware). The server-side modules that provide back-end functionality also include a search module 207 which provides a search interface for searching the objects in the Application Entitlement Store (AES) Object Store 214. In an example embodiment, the objects in the AES Object Store 214 might be indexed in a Solr index in preparation for searching. Solr is an open-source enterprise search platform from the Apache Lucene project.

[0055] Another server-side module that provides back-end functionality is module 208 for user reports. Module 208 creates reports for administrators. In an example embodiment, module 208 will retrieve data for the reports from Logging/Analytics Store 215 and AES Object Store 214, among other places. The reports might cover licenses, application usage, access location (e.g., geolocation), devices by user and/or type and/or configuration, etc. In an example embodiment, the reports from Logging/Analytics Store 215 might be created using Apache's Hadoop (e.g., Hadoop jobs). Also in an example embodiment, the reports might generate cubes, such as OLAP (online analytical processing) cubes, where the cubes might have dimensions for, among other things: (a) user; (b) group/tag; (c) application; (d) date/time; and (e) device type. It will be appreciated that each of these dimensions might not be generated in the same cube for the same report. In an example embodiment, a user report might also include metrics related to the number of application provisions, usage time, usage time at a location, number of logins, etc.

[0056] The server-side modules that provide back-end functionality include module 209 for provisioning. In an example embodiment, module 209 might have custom logic (or provisioning drivers) for different types of applications. For example, for SaaS applications 220, the provisioning driver might operate according to the process defined in Table 1.

TABLE 1

If the SaaS application supports JIT (Just In Time) provisioning,
Then do nothing.
If the SaaS supports Service Provisioning Markup Language (SPML) or another provisioning API,
Then call that API.
Otherwise, do form stuffing to automatically provision the SaaS application.

[0057] For example, since Salesforce's SaaS employs JIT provisioning, when a user attempts to access the Salesforce SaaS application, Salesforce will contact the system for an immediate provisioning decision.

[0058] Form stuffing is used for authentication for SaaS applications 220 when the SaaS provider does not support SAML (or another SSO scheme). That is to say, module 209 sets a password during provisioning. In an example embodiment, the passwords might be randomly generated, so that compromising a single password will not compromise the system. The user may be prevented from accessing the password. By not providing the password to the user, the system prevents the user from going to a SaaS application without using a system agent.

[0059] For thin applications and/or native applications, the provisioning driver might (1) perform any licensing accounting and (2) make the application available for download. For remote applications, the provisioning driver might (1) perform any licensing accounting and (2) retrieve related files, e.g., Remote Desktop Protocol (RDP) files, VMware View™ files, etc. In an alternative example embodiment, some or all of these operations might be performed by existing workflow and licensing processes (e.g., in response to provisioning callouts as described above).

[0060] It will be appreciated that in an example embodiment, the thin application might be a virtualized application such as a VMware ThinApp™, which provides a sandbox to hold runtime modifications to, for example, the registry and/or file system made by an application. It will further be appreciated that such a sandbox is useful to prevent overwritten DLLs (dynamically linked libraries), among other things. Similarly, in an example embodiment, the remote application might be a desktop virtualization application such as a VMware View application which runs on a guest operating system on a virtual machine provided by a virtualization layer (e.g., a hypervisor) on a remote server.

[0061] Further, in an example embodiment, the provisioning driver for an application might support the following operations: (1) add/remove a user; and (2) get user status. Typically, the add/remove operation will be performed once per user. If the application is a SaaS application, the provisioning driver might add a user by associating the user's credentials with an account previously obtained by an administrator from a SaaS provider. If

the application is a thin application or a remote application, the provisioning driver might add the user by performing any licensing accounting.

[0062] It will be appreciated that the retrieved user status might be used to determine variances between the expected provision state and the actual provision state of a user of a SaaS application.

[0063] As also depicted in Figure 2, the server-side modules that provide back-end functionality include module 210 for entitlements. In an example embodiment, this module controls access to the system's applications based on entitlements. In an example embodiment, an entitlement might be represented by as a tuple with elements such as <user, device, application, set of user tags, device configuration, authentication type>.

[0064] Among other things, the entitlements determine which application shortcuts (e.g., as represented by icons) are presented on the GUI desktop displayed by native client 103. Similarly, these entitlements will be checked to determine which application shortcuts (e.g., as represented by icons) are presented in the GUI displayed by User Web UI 202. Additionally, for each SSO operation against an application, module 210 checks the entitlement to verify that the SSO can proceed and to set a session timeout. When entitlements change, module 210 determines whether any provisioning/de-provisioning operations should be performed and whether any SSO sessions should be closed.

[0065] Another server-side module that provides back-end functionality is module 211 for SSO. In an example embodiment, module 211 keeps track of authentication sessions that are shared across different applications. It will be appreciated that initiation of a session involves user authentication. In an example embodiment, if the user is inside the firewall 104, authentication might take place against AD Server 106 (or other directory service), by going through Enterprise Connector 105. If the user is outside firewall 104, authentication might take place against AD Server 106 (or other directory service), by going through both Application Manager 102 and Enterprise Connector 105. Alternatively, the system might delegate authentication to an SaaS provider (e.g., Google Apps) or other provider (e.g., Yahoo!) that provides hosted identity and which is accessed using SAML or OAuth (Open Authorization).

[0066] In other instances, authentication might not involve a directory. That is to say, authentication might occur through SAML. Or for down-level applications, authentication

might be handled by form stuffing as described above, which might request a username and/or password from a user through a GUI in web browser 103d.

[0067] The server-side modules that provide back-end functionality also include module 212 for user update. This module updates user data based on changes, for example: (1) resulting from operations performed through Administrator Web UI 201; (2) coming from RESTful API 203; or (3) propagated from Directory Store 219 through a synchronization tool, e.g., an AD update tool, a Yahoo! update tool, a Google update tool, etc.

[0068] Another server-side module that provides back-end functionality is Billing Service 213, which is might be an automated third-party service that performs the accounting or verification of accounting for licensing charges/payments for applications provisioned by the system. In an example embodiment, the system might access Billing Service 213 using an API.

[0069] The server-side modules that provide back-end functionality also include AES Object Store 214. In example embodiment, AES Object Store 214 is a scalable object store that stores the system's objects (e.g., related to users, applications, groups, application families, entitlements, account activations, etc.) in a database. It will be appreciated that the rest of the system works with versions of these objects using transactions semantics. In an example embodiment, AES Object Store 214 provides optimistic concurrency and atomic operations. Also, in an example embodiment, AES Object Store 214 might be distributed among several databases on the basis of enterprise organizational structure.

[0070] Another server-side module that provides back-end functionality is Logging/Analytics Store 215. In an example embodiment, Logging/Analytics Store 215 might provide scalable persistent storage (e.g., in the form of logs) for: (1) API calls; (2) provisioning operations; and (3) application usage data (e.g., received from a native client or browser plug-in). Logging/Analytics Store 215 also provides a job interface which allows User Reports 208 to run analytics on the logs and generate reports and cubes.

[0071] The server-side modules that provide back-end functionality also include BLOB Store 216, Key Chain Store 217, and User Key Store 218. BLOB Store 216 stores images for native applications, thin applications, etc. As indicated in Figure 2, BLOB Store 216 might be provided by a third party. In an example embodiment, the system might access BLOB Store 216 using an API. Key Chain Store 217 stores the encrypted passwords used in

down-level authentication. And User Key Store 218 stores the per-user keys used to decrypt the encrypted passwords in Key Chain Store 217.

[0072] It will be appreciated that in an example embodiment, the server-side modules described above might be part of the Application Manager 101 described. In an alternative example embodiment, some of the server-side modules described above might instead be part of the Enterprise Connector 105, if it is installed. Or the functionality in the server-side modules might be relocated to client-side modules and vice versa. That is to say, the modules described above can be distributed throughout the system in any suitable way.

TABLE 2

	Type 1 Enterprise	**Type 1 (Variant)**	**Type 2 Enterprise**
Modern SaaS App	User inside firewall 2 STSes SAML to App	User outside firewall 1 STSes SAML to App	No firewall 1 STS Proprietary connector to Id Store SAML to App
Legacy SaaS App	1 STS HTTP Fed to App	No STS Native auth to store HTTP Fed to App	No STS Native auth to store HTTP Fed to App

[0073] Table 2 illustrates various scenarios for SSO, in accordance with an example embodiment. It will be appreciated that these scenarios are intended to illustrative rather than limiting. In the first scenario in the column Type 1 Enterprise and in the row Modern SaaS App, the user is inside the firewall but there are two secure token services (STSs), e.g., the application manager server does not have a secure connection into the firewall. This scenario is depicted in Figure 3A, where STS 302 is onsite (e.g., Active Directory Federation Service (ADFS) in the DMZ) and can authenticate the user to an AD server and create a SAML token. This is shown as operation 303 in Figure 3B, between client device 301 and STS 302. The other STS, STS 304, can create a SAML token based on an open session using the SAML token created by STS 302. This is shown in operation 305 in Figure 3B, between client device 301 and application portal 306 (e.g., the user web portal described above). As depicted in operation 307 in that figure, the client device 301 then uses the SAML token created by STS 304 to connect to a modern (SAML-enabled) SaaS application server 308 and gain access to an SaaS application.

[0074] In the second scenario, described in the column Type 1 Enterprise and the row Legacy SaaS App of Table 2, there is only one STS. For example, the application

manager server may have a secure connection into the firewall. The STS is on an application manager server, which can authenticate the user to an AD server and which proxies a connection between the client device and the legacy SaaS application server using HTTP Federation (HTTP Fed). It will be appreciated that in this instance, the application manager server performs authentication on behalf of the user using a traditional username-password combination. In an alternative example embodiment, the system might also use HTTP Fed with an SAML-enabled SaaS application server, in any of the scenarios described in the table.

[0075] In the third scenario, described in the column Type 1 Variant and in the row Modern SaaS App of Table 2, the user is outside the firewall and there is only one STS, e.g., the application manager server has a secure connection into the firewall. This scenario is depicted in Figure 3B, wherein an application manager server includes STS 304 which can authenticate the user to an AD server 309 through the use of directory connector 310 (e.g., AD Connector in the DMZ), such as the Enterprise Connector 105 described above, and create a SAML token. This is shown in operation 305, between client device 301 and application portal 306 (e.g., the user web portal described above), and operation 311, between STS 304 and directory connector 310. Then as depicted in operation 307 in that figure, the client device uses that SAML token to connect to a SAML-enabled SaaS application server 308 and gain access to an SaaS application.

[0076] In the fourth scenario, described in the column Type 1 (Variant) and in the row Legacy SaaS App of Table 2, there is no STS and a native authorization is made to the application manager. The authentication manager can then authenticate the user to an identity store on premises and which proxies a connection between the client device and the legacy SaaS application server using HTTP Federation (HTTP Fed). As indicated in the table, this same scenario occurs in cell 6 in the column Type 2 Enterprise and the row Legacy SaaS App.

[0077] In the fifth scenario, described in the column Type 2 Enterprise and in the row Modern SaaS App of Table 2, there is no firewall and one STS. This scenario is depicted in Figure 3C, wherein an application manager server includes STS 304 which can authenticate the user to an identity store and create a SAML token from a SAML token created by a hosted identity store (e.g., at Google or Salesforce) accessed using a proprietary connector. This is shown in operation 305, between client device 301 and application portal 306 (e.g., the user web portal described above), and operation 314, between application manager 312 and hosted-identity manager 313. The client device 301 then uses that SAML token to

connect to a SAML-enabled SaaS application server 308 and gain access to a SaaS application, in operation 307.

[0078] Figure 4 is a sequence diagram showing a process for entitling and provisioning a new SaaS account, in accordance with an example embodiment. It will be recalled that an entitlement is a permission that associates a particular user, or group of users, to a particular application, or family of applications. When the entitlement is for an SaaS application, whether it is hosted by a 3rd party or within the enterprise's own network, an account is created for each user with the SaaS provider when the entitlement is created. As indicated in the figure, this process might be performed by several of the modules depicted in Figure 2, in an example embodiment.

[0079] In operation 401 of the process, the system receives a call from the administrator web portal (ADMIN UI) to add an entitlement to a SaaS application for a user or group of users, e.g., as the result of a GUI command from an administrator. This call goes to the system's Entitlement module. This call is logged to the Logging/Analytics Store as are any other calls that make use of the Internal Java API. Then in operation 402, the Entitlement module adds the entitlement and expected provisioning state to the AES Object Store's objects for the affected users. In operation 403, the Entitlement module makes a call (e.g., through the Internal Java API) to the Provisioning module to provision any accounts that need to be provisioned. And in operation 404, the Provisioning module makes calls (e.g., through a provisioning API) to the 3rd Party SaaS Provider (or Vendor) to provision SaaS accounts for the user or group of users. In operation 405, the Provisioning module updates the pertinent objects in the AES Object Store with the state of the provision and, if appropriate, modifies the number of licenses for the SaaS application. Then in operation 406, the Provisioning module updates the Logging/Analytic Store with the progress (or failure) of the provision. In operation 407, the user web portal (User Web UI) calls the Entitlement module (e.g., through the Internal Java API) to determine whether there are any new entitlements for the user, e.g., as the result of a GUI command from the user (or one of the users in the group of users). And the Entitlement module causes the user web portal to display the added entitlement. Alternatively, in operation 408, a Native Client (e.g., running on a device such as an iPad), calls the Entitlement module (e.g., through the Internal Java API) to determine whether there are any new entitlements for the user, e.g., as the result of a GUI command from the user (or one of the users in the group of users). And the Entitlement module causes the Native Client to display the added entitlement.

[0080] Figure 5 is a flowchart diagram that illustrates a process for storing provisioning tasks, in accordance with an example embodiment. In an example embodiment, the operations in this process might be performed by the system's Entitlement module. The process begins in operation 501 with operations to create an entitlement to an application (e.g., a thin application, a remote application, or a SaaS application) for a user or group of users. In operation 502, the Entitlement module determines which provisioning tasks (or operations) should be performed, e.g., modifying a license number. Then in operation 503, the Entitlement module adds (1) the entitlement and (2) the provisioning tasks to the AES Object Store or other database. In operation 504, the Entitlement module notifies the Provisioning module about the new provisioning tasks, e.g. through the Internal Java API.

[0081] Figure 6 is a flowchart diagram that illustrates a process for determining provisioning tasks, in accordance with an example embodiment. In an example embodiment, the operations in this process might be performed by the system's Entitlement module. The process begins in operation 601 with the receipt of a tuple, e.g., <Application or Application Family, User/Group/Expression, Device Policy>. The expression allows an administrator to exclude users and user groups from policies and to apply policies to multiple groups or partial groups. In operation 602, the Entitlement module determines what the entitlement is for. If the entitlement is for an application, the Entitlement module creates an application list with the application in it, in operation 603. If the entitlement is for an application family, the Entitlement module creates an application list with the closure (or mathematical closure) of the applications in the application family, in operation 604. In operation 605, the Entitlement module determines who the entitlement is for. If the entitlement is for a single user, the Entitlement module creates a user list with a single user, in operation 606. If the entitlement is for a group or expression, the Entitlement module creates a user list with a closure (or mathematical closure) of everyone in the tuple element, user/groups/expression, in operation 607. Then in operation 608, the Entitlement module generates provisioning tasks (or operations) from the cross product of the application list and the user list. In this example, all provisioning tasks (or operations) have the same device policy. In operation 609, the Entitlement module returns a list of provisioning tasks (or operations) in the form of a tuple, e.g., <user, application, device policy>.

[0082] Figure 7 is a flowchart diagram that illustrates a process for provisioning an application, in accordance with an example embodiment. In an example embodiment, the operations in this process might be performed by the system's Provisioning module. In

operations 701, the provisioning module determines whether there are any provisioning tasks (or operations) to process in the database (or the AES Object Store 713) where the provisioning tasks are stored, e.g., by the Entitlement module. If not, the Provisioning module remains in a busy loop, waiting for such a task. If there is a provisioning task to process, the Provisioning module determines which provisioning driver to use, based on the application type (e.g., thin application, remote application, SaaS application, etc.) as indicated in the provisioning task, in operation 702. Then, in operation 703, the Provisioning module determines whether a callout flag is set, e.g., whether the provisioning task uses provisioning callouts, e.g., that integrate with existing workflow and licensing processes. If a callout flag is set, the provisioning module determines whether the Enterprise Connector is installed (or set up), in operation 704. If the Enterprise Connector is installed, the Provisioning module issues the provisioning callouts through the Enterprise Connector, in operation 705. If the Enterprise Connector is not installed, the Provisioning module issues the provisioning callouts itself, in operation 706. In operation 707, the Provisioning module determines whether the provisioning callouts were successfully completed, e.g., by the existing workflow and licensing processes. If not, the Provisioning module logs the error, in operation 708, and returns to the beginning of the process, operation 701.

[0083] If the provisioning callouts were successful or if provisioning callouts were not used, the Provisioning module calls the applicable provisioning driver, using the provisioning task data, in operation 709. Then, in operation 710, the Provisioning module determines whether the call (or provision) was successful. If not, the Provisioning module updates the provisioning object in the AES Object Store 713 to indicate the failure, in operation 711, and then logs the error and returns to operation 701. If the call to the provisioning driver was successful, the Provisioning module updates the provisioning object in the AES Object Store 713 to indicate completion, in operation 712, and then returns to operation 701.

[0084] Figure 8 is a flowchart diagram that illustrates a process carried out by a provisioning driver for provisioning an application that does not support SAML, in accordance with an example embodiment. This example application can be accessed and used through a local client rather than a web browser. Since the SaaS application does not support SAML, the system creates a random password, encrypts, and saves it in operation 801. This password can then be used for SSO to the application, though the user will never see it. This also makes the password uncorrelated with any other passwords, so cracking that

password will not allow access to anything other than the particular SaaS application for the particular user's account.

[0085] In operation 802, the provisioning driver makes a call to the SaaS application (e.g., through an API) to create an account for the user. The provisioning driver creates an SaaS account in the AES Object Store 808 for the Mozy account, in operation 803. And in operation 804, the provisioning driver updates the SSO service (e.g., through a call using the Internal Java API), with a tuple such as <username, password, name of SaaS application>, where the username is the username for the SaaS application account and the password is the password for the SaaS application account, e.g., the random password. In operation 805, the SSO service updates the KeyChain Store with the password. The provisioning driver provisions the user to the SaaS application under the default group, in operation 806. And in operation 807, the provisioning driver creates a SaaS account in the AES Object Store with the new provision key resulting from that provision.

[0086] Figure 9A and 9B are flowchart diagrams that illustrate a process for authenticating a user to a SaaS application from a session, in accordance with an example embodiment. When a user accesses the system to start an SSO session to access and use a SaaS application, the system will try, in the first instance, to create a token for a SaaS application, based on an existing session. Thus, in operation 901 in Figure 9A, the system looks up the session and, in operation 902, determines whether the session is valid. If the session is not valid, the system redirects to the SaaS service (e.g., the Application Manager) with an error message. If the session is valid, the system determines whether the session is alive, in operation 903. If not, the system creates a session object with any device information in the initial SSO request, in operation 904, and redirects to the login page, in operation 905, for a credentialed login with a username and password. If the session is alive, the system looks up the user's entitlement for the SaaS application, in operation 906, and ascertains any device or application policies from the entitlement, in operation 907. In operation 908, the system determines whether the current session satisfies those policies. If not, the system redirects to the SaaS service (e.g., the Application Manager) with an error message. If the current session satisfies the policies, the system redirects to the SaaS service (e.g., the Application Manager) with a valid session token, in operation 910 in Figure 9A.

[0087] As indicated above, the system will redirect to a login page in operation 904, if the session is not alive. In an example embodiment, this redirect might occur via the SAML SP (Service Provider) Initiated SSO profile (with the Redirect Bindings). If an Enterprise

Connector is installed with the system, the redirect will go to the Enterprise Connector which will now act as an IdP against the internal LDAP store (e.g., AD). The browser on the client device will negotiate an authentication protocol (e.g., Kerberos) and authenticate the user. The Enterprise Connector will then redirect the authenticated user back to the Application Manager via the response portion of SAML SP Initiated SSO profile (over the Redirect Bindings). The Application Manager will then create the SSO session for the SaaS application.

[0088] Figure 9B picks up with the system receiving an authentication response from the IdP (Identity Provider), in operation 911. In operation 912, the system looks up the user/device session for the authentication response. If the authentication is unsuccessful, the system adds the failure to the session, in operation 913, and redirects to the initiating SaaS service (e.g., the Application Manager) with an error message, in operation 914. If the authentication is successful, the system adds any session expiration data from the SAML SSO response to the session, in operation 915. Then in operation 916, the system adds any assertions from the SAML SSO response to the session. And in operation 917, the system adds any attributes from the SAML SSO response to the session. In operation 918, the system completes the authentication of the user to the SaaS application provider from a session.

[0089] Figure 10 shows a GUI view of a dashboard tab displayed by an administrator web portal, in accordance with an example embodiment. As depicted in this figure, a GUI view in an administrator portal might consist of multiple tabs for, e.g., Dashboard, Users, Group, Applications, Licenses, and Policy. Among other things, the Dashboard tab 1001 displays counts for Users Under Management, SaaS Applications Under Management, Desktop Applications Under Management, and GB of Storage Utilized. It will be recalled that the desktop applications might include thin applications such as applications virtualized using VMware ThinApp™ and remote applications accessed by VMware View™, in an example embodiment. The Dashboard tab also displays a list of reports 1002 generated by the system, including reports on: (1) Most active applications; (2) Application usage over time; (3) Most active users; (4) Most active locations (e.g., geolocations); (5) Most recent locations (e.g., geolocations); (6) System utilization; (7) Cost of license under-usage; (8) Cost of license over-subscription; (9) Cost of licenses (table); and (10) cost of licenses (chart).

[0090] Figure 11 shows a GUI view of a users tab displayed by an administrator web portal, in accordance with an example embodiment. Among other things, the Users tab 1101

displays a table of data regarding the users managed by the system. The table's columns include: (1) name; (2) user name; (3) groups; and (4) role. The Users tab 1101 also displays the reports 1102 related to users and includes a GUI control (e.g., a button) 1103 that allows an administrator to add a user.

[0091] Figure 12 shows a GUI view of a user edit dialog displayed by an administrator web portal, in accordance with an example embodiment. In an example embodiment, this dialog might be displayed in the Users tab 1101, e.g., when an administrator clicks on a user name in the Users tab 1101. The user edit dialog 1201 includes edit boxes 1202 that allow for editing of a user's first and last name and a check box 1203 that allows the user to be made an administrator. The user edit dialog 1201 also includes check boxes 1204 that allow the user to be made a part of groups such as sales and finance.

[0092] Additionally, in an example embodiment, the user edit dialog 1201 lists in a table 1205 the applications to which the user is entitled. The table 1205 has columns for the application name, device (e.g., client device) policy, and deployment. Thus, for example, the user whose name is Regis Gorman is entitled to use the Salesforce application on all devices and deployment is automatic. The user edit dialog 1201 also includes a GUI control (e.g., a button) 1206 to add an entitlement and a GUI control (e.g., a button) 1207 to deny an entitlement. And the user edit dialog 1201 also includes a GUI control (e.g., a button) 1208 to remove an entitlement to an application and a GUI control (e.g., a button) 1209 to save editing changes.

[0093] Figure 13 shows a GUI view of a groups tab displayed by an administrator web portal, in accordance with an example embodiment. Among other things, the Groups tab 1301 displays a table of data regarding the groups managed by the system. The table's columns include: (1) name; (2) number of users; and (3) number of application entitlements. The Groups tab 1301 also displays the reports 1302 (e.g., most active users, most active locations, and most recent locations) related to groups and includes a GUI control (e.g., a button) 1303 that allows an administrator to add a group.

[0094] Figure 14 shows a GUI view of a group edit dialog displayed by an administrator web portal, in accordance with an example embodiment. In an example embodiment, this dialog might be displayed in the Groups tab 1301, e.g., when an administrator clicks on a group name in the Groups tab 1301. The group edit dialog 1401 includes edit box 1402 that allows for editing of a group's name. The group edit dialog 1401

also includes a GUI control (e.g., a button) 1403 that allows a user to be added to a group and a GUI control (e.g., a button) 1404 that allows a user to be removed from a group.

[0095] Additionally, in an example embodiment, the group edit dialog 1401 lists in a table 1405 the applications to which a group is entitled. The table 1405 has columns for the application name, device (e.g., client device) policy, and deployment. Thus, for example, the group whose name is Company X is entitled to use the Salesforce application on all devices and deployment is automatic. The group edit dialog 1401 also includes a GUI control (e.g., a tree view) 1406 with a scroll bar 1407 to add an entitlement for the group, e.g., by clicking on the application's icon. And the user edit dialog 1401 also includes GUI control 1408 that allows an entitlement for the group to be edited and a GUI control (e.g., a button) 1409 to remove an entitlement for the group. The user edit dialog 1401 includes GUI control (e.g., a button) 1410 to save editing changes.

[0096] Figure 15 shows a GUI view of an entitlement edit dialog displayed by an administrator web portal, in accordance with an example embodiment. In an example embodiment, this dialog might be displayed when an administrator clicks on a GUI control (e.g., GUI control 1408) in group edit dialog 1401. The entitlement edit dialog 1501 allows for editing of the "Company X" group's entitlement for application Box.net. The edit dialog 1501 includes dropdown list 1502 that allows for changing the devices (e.g., client devices) encompassed within the entitlement and a dropdown list 1503 that allows for changing the deployment of the entitlement. The entitlement edit dialog 1501 also includes a GUI control (e.g., a button) 1504 to save editing changes.

[0097] Figure 16 shows a GUI view of an applications tab displayed by an administrator web portal, in accordance with an example embodiment. Among other things, the Applications tab 1601 displays a table of data regarding the applications managed by the system. The table's columns include: (1) name and description; (2) type (e.g., remote application, native, Google Apps, etc.); and (3) licensing requirement and/or use. The Applications tab 1601 also displays the reports 1602 related to applications and includes a GUI control (e.g., a button) 1603 that allows an administrator to add an application. The Applications tab 1601 also includes a text field 1604 that displays user tips.

[0098] Figure 17 shows a GUI view of a dialog for adding an application, in accordance with an example embodiment. In an example embodiment, this dialog might be displayed in the Applications tab 1601 of an administrator web portal, e.g., when an

administrator clicks on GUI control (e.g., a button) 1603 to add an application. The dialog 1701 includes an edit box 1702 for the name of the application and edit box 1703 for a description of the application. The dialog 1701 also includes a checkbox 1704 to indicate whether a license is required.

[0099] Additionally, the dialog 1701 includes a list 1705 of radio buttons that identify the application type for the application: (1) pre-configured applications, including email and data sync services; (2) Windows desktop applications, including VMware ThinApp virtualized applications or native MSI (Windows Installer) applications; (3) SaaS applications that are web-based applications like Workday or Salesforce; and (4) published (or remote) applications such as the Microsoft Terminal Services (RDS) application or the Citrix XenApp application. It will be appreciated that as depicted in Figure 17, the top radio button in list 1705 has been checked. And consequently, dialog 1701 displays the icons 1706 for four pre-configured applications managed by the system: (A) VMware View; (B) Mozy; (C) Google Apps; (D) Box.net; and (E) Salesforce. Dialog 1701 also includes a GUI control (e.g., a button) 1707 to save editing changes.

[00100] Figure 18 shows a GUI view of an edit dialog for editing a Windows desktop application, in accordance with an example embodiment. In an example embodiment, this dialog might be displayed in the Applications tab 1601 of an administrator web portal, e.g., when an administrator clicks on an application icon displayed when a user selects the radio button for Windows desktop application in list 1705 of radio buttons in dialog 1701. The edit dialog 1801 includes an edit box 1802 which allows the name of the Windows desktop application to be edited and an edit box 1803 which allows the description of the Windows desktop application to be edited. The edit dialog 1801 also includes a checkbox 1804 to indicate whether a license is required. And the edit dialog 1801includes a GUI control (e.g., a button) 1805 to add the Windows desktop application to an application family and a GUI control (e.g., a button) 1806 to edit application type.

[00101] Additionally, edit dialog 1801 includes a GUI control (e.g., a button) 1807 that allows the administrator to select a file to be uploaded by a local host computer, e.g., the host computer of a user entitled to use the Windows desktop application. After clicking on GUI control 1807, pop-up dialog 1808 is displayed, which allows the administrator to select the file using a mouse and/or keyboard from a list of files 1809 in a directory. Pop-up dialog 1808 also allows the administrator to select a location from a list of locations (e.g., DEVICES, SHARED, and PLACES) 1810, using a mouse and/or keyboard. And pop-up

dialog 1808 includes GUI control (e.g., a button) that allows for cancellation of the file selection operation.

[00102] Figure 19 shows a GUI view of an edit dialog for editing a published (or remote) application, in accordance with an example embodiment. In an example embodiment, this dialog might be displayed in the Applications tab 1601 of an administrator web portal, e.g., when an administrator clicks on an application icon displayed when a user selects the radio button for published application in list 1705 of radio buttons in dialog 1701. The edit dialog 1901 includes an edit box 1902 which allows the name of the published application to be edited and an edit box 1903 which allows the description of the Windows desktop application to be edited. The edit dialog 1901 also includes a checkbox 1904 to indicate whether a license is required. And the edit dialog 1901 includes a GUI control (e.g., a button) 1905 to add the published application to an application family and a GUI control (e.g., a button) 1906 to edit application type. Additionally, edit dialog 1901 includes an edit box 1907 for the program name, edit box 1908 for the server identifier, and edit box 1909 for the server port.

[00103] Figure 20 shows a GUI view of a licenses tab displayed by an administrator web portal, in accordance with an example embodiment. Among other things, the Licenses tab 2001 displays a table of data regarding the licenses managed by the application-management system. The table's columns include: (1) the applications licensed; (2) quantity of licenses; and (3) the cost per license. The licenses tab 2001 also displays the reports 2002 related to the licenses.

[00104] Figure 21 shows a GUI view of a policy tab displayed by an administrator web portal, in accordance with an example embodiment. Among other things, the policy tab 2101 allows an administrator to set polices relating to client devices which, in turn, will determine application entitlement. The policy tab 2101 displays three checkboxes and an edit box for setting a policy for remote access by a client device. If checked, checkbox 2102 indicates that the client device must be corporate owned on an AD domain. It will be appreciated that such a policy might be applicable to an application that accesses sensitive records, such as personnel records or medical records. The related edit box 2103 allows the AD domain to be identified. If checked, checkbox 2104 indicates that virus protection must be installed on the client device. And if checked, checkbox 2105 indicates that updates must be automatically installed on the client device (e.g., Windows operating system updates, virus definition updates, etc.).

[00105] This policy is intended to be illustrative and not limiting. Another device policy might require that Windows be installed. Or the device policy might pertain to the channel (or connection) used to access the AD domain, e.g., LAN, WAN, name-password protected, secure-ID protected, etc. Further still, the device policy might pertain to geolocation, e.g., an application can only be used on a client device if the client device is located within the jurisdictional boundaries of a particular governmental entity.

[00106] Additionally, policy tab 2101 includes a GUI control (e.g., a button) 2106 that allows an administrator to add a new policy and a text field 2107 that displays user tips. And policy tab 2101 includes a GUI control (e.g., a button) 2108 to save editing changes.

[00107] Figure 22 shows a GUI view displayed by a user web portal, in accordance with an example embodiment. In an example embodiment, this view might be displayed by the system after a user (e.g., Regis Gorman) logs in. View 2201 lists icons of the applications that the user is entitled to and that have been enabled. If a user clicks on an icon, the application will launch in a relatively seamless manner.

[00108] It will be appreciated that GUI view 2201 is intended to be illustrative rather than limiting. If the client device is a Windows device, GUI view 2201 might also include icons added to the Start menu and status alerts as to the applications in the status bar, in order to provide the user with a sense of seamlessness. Of course, enhancements of a similar nature might also be made to GUI view 2201 when displayed on an Apple device, to promote seamlessness.

[00109] Figure 23 shows another GUI view displayed by a user web portal, in accordance with an example embodiment. In an example embodiment, this view might be displayed by the application-manager system after a user (e.g., Regis Gorman) logs in. View 2301 lists icons of the applications that the user is entitled to and that have been enabled, e.g., Salesforce, as well as the applications that the user is entitled to, but has not enabled, e.g., Workday. In an example embodiment, the user can install an application by clicking on install button.

[00110] Any of the operations described herein that form part of the inventions are useful machine operations. The inventions also relate to a device or an apparatus for performing these operations. The apparatus may be specially constructed for the required purposes, such as the carrier network discussed above, or it may be a general purpose computer selectively activated or configured by a computer program stored in the computer.

In particular, various general purpose machines may be used with computer programs written in accordance with the teachings herein, or it may be more convenient to construct a more specialized apparatus to perform the required operations.

[00111] The inventions can also be embodied as computer readable code on a computer readable medium. The computer readable medium is any data storage device that can store data, which can thereafter be read by a computer system. Examples of the computer readable medium include hard drives, network attached storage (NAS), read-only memory, random-access memory, CD-ROMs, CD-Rs, CD-RWs, DVDs, Flash, magnetic tapes, and other optical and non-optical data storage devices. The computer readable medium can also be distributed over a network coupled computer systems so that the computer readable code is stored and executed in a distributed fashion.

[00112] Although example embodiments have been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications can be practiced within the scope of the following claims. For example, the processes described herein might be used to provision Platform as a Service (PaaS) or Infrastructure as a Service (IaaS), in addition to Software as a Service (SaaS), thin applications, remote applications, etc. Further, the process operations described above can be ordered, modularized, and/or distributed in any suitable way.

[00113] Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the inventions are not to be limited to the details given herein, but may be modified within the scope and equivalents of the following claims. In the following claims, elements and/or steps do not imply any particular order of operation, unless explicitly stated in the claims or implicitly required by the disclosure.

What is claimed is:

CLAIMS

1. A method for providing applications to a user, the method comprising operations of:

authenticating a user of a client device over a channel, wherein the authentication operation is performed using a directory service;

presenting a plurality of applications in a graphical user interface (GUI) displayed by the client device, wherein the plurality of applications depends at least in part on the authentication, the client device, and the channel and wherein the plurality of applications includes a thin application and a software-as-a-service (SaaS) application;

receiving a selection as to an application from the user;

provisioning the SaaS application, if the selection is for the SaaS application, wherein the provisioning includes automatically logging the user onto an account with a provider of the SaaS application using a single sign-on protocol and connecting the user to the account so that the user can interact with the SaaS application; and

launching the thin application, if the selection is for the thin application, wherein each operation of the method is executed by one or more processors.

2. The method of claim 1, wherein the thin application is a sandboxed application.

3. The method of claim 1, wherein the plurality of applications further includes a remote application.

4. The method of claim 3, wherein the remote application is a desktop virtualization application.

5. The method of claim 1, wherein the SaaS application is a web application.

6. The method of claim 1, wherein the plurality of applications displayed to the user depends at least in part on the geolocation of the user.

7. The method of claim 6, further comprising the operation of gathering statistics on use of the plurality of applications displayed to the user and the geolocation of the use.

8. The method of claim 1, wherein the provisioning operation is performed just-in-time (JIT).

9. The method of claim 1, wherein the directory service is an Active Directory (AD) service.

10. A computer-readable storage medium, which is non-transitory, storing a program that when executed, instructs one or more processors to perform the following operations:

authenticate a user of a client device over a channel, wherein the authentication operation is performed using a directory service;

presenting a plurality of applications in a graphical user interface (GUI) displayed by the client device, wherein the plurality of applications depends at least in part on the authentication, the client device, and the channel and wherein the plurality of applications includes a thin application and a software-as-a-service (SaaS) application;

receiving a selection as to an application from the user;

provisioning the SaaS application, if the selection is for the SaaS application, wherein the provisioning includes automatically logging the user onto an account with a provider of the SaaS application using a single sign-on and connecting the user to the account so that the user can interact with the SaaS application; and

launching the application, if the selection is for the thin application.

11. The computer-readable storage medium of claim 10, wherein the thin application is a sandboxed application.

12. The computer-readable storage medium of claim 10, wherein the plurality of applications further includes a remote application.

13. computer-readable storage medium of claim 12, wherein the remote application is a desktop virtualization application.

14. The computer-readable storage medium of claim 10, wherein the SaaS application is a web application.

15. The computer-readable storage medium of claim 10, wherein the plurality of applications displayed to the user depends at least in part on the geolocation of the user.

16. The computer-readable storage medium of claim 15, further comprising the operation of gathering statistics on use of the plurality of applications displayed to the user and the geolocation of the use.

17. The computer-readable storage medium of claim 10, wherein the provisioning operation is performed just-in-time (JIT).

18. The computer-readable storage medium of claim 10, wherein the directory service is an Active Directory (AD) service.

19. A method for providing applications to a user, the method comprising operations of:

authenticating a user of a client device over a channel, wherein the authentication operation is performed using a directory service;

presenting a plurality of applications in a graphical user interface (GUI) displayed by the client device, wherein the plurality of applications depends at least in part on the user

authentication, the client device, and a geolocation of the client device and wherein the plurality of applications includes a sandboxed application and a software-as-a-service (SaaS) application;

receiving a selection as to an application from the user;

provisioning the SaaS application, if the selection is for the SaaS application, wherein the provisioning includes automatically logging the user onto an account with a provider of the SaaS application using a single sign-on and connecting the user to the account so that the user can interact with the SaaS application; and

launching the sandboxed application, if the selection is for the sandboxed application, wherein each operation of the method is executed by one or more processors.

20. The method of claim 20, wherein the plurality of applications displayed to the user further includes a virtualized desktop application.

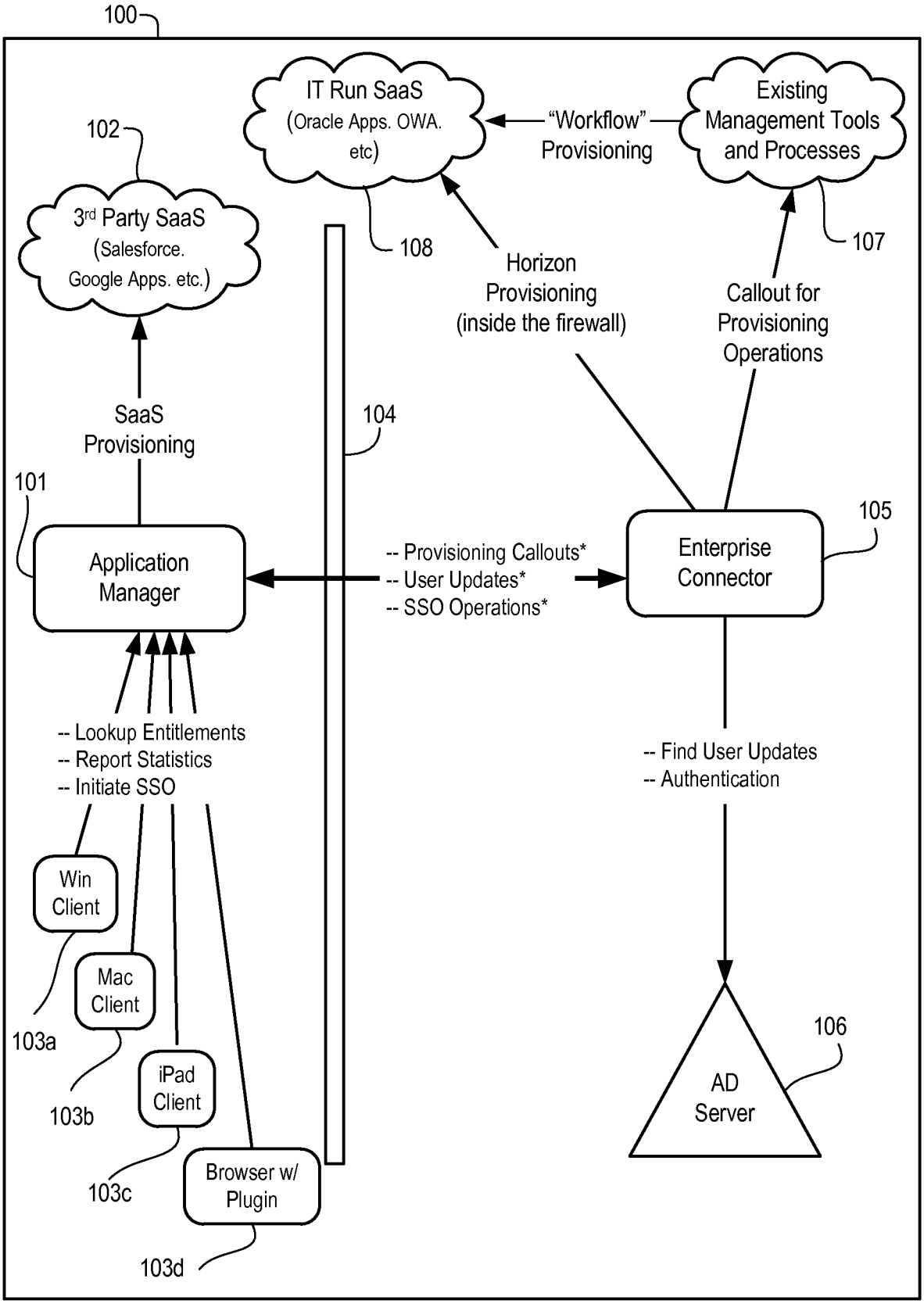
100
IT Run SaaS
(Oracle Apps. OWA. etc)
"Workflow" Provisioning
Existing Management Tools and Processes
102
3rd Party SaaS
(Salesforce. Google Apps. etc.)
108
107
Horizon Provisioning
(inside the firewall)
Callout for Provisioning Operations
SaaS Provisioning
104
101
Application Manager
-- Provisioning Callouts*
-- User Updates*
-- SSO Operations*
Enterprise Connector
105
-- Lookup Entitlements
-- Report Statistics
-- Initiate SSO
-- Find User Updates
-- Authentication
Win Client
Mac Client
iPad Client
Browser w/ Plugin
103a
103b
103c
103d
AD Server
106

Fig. 1

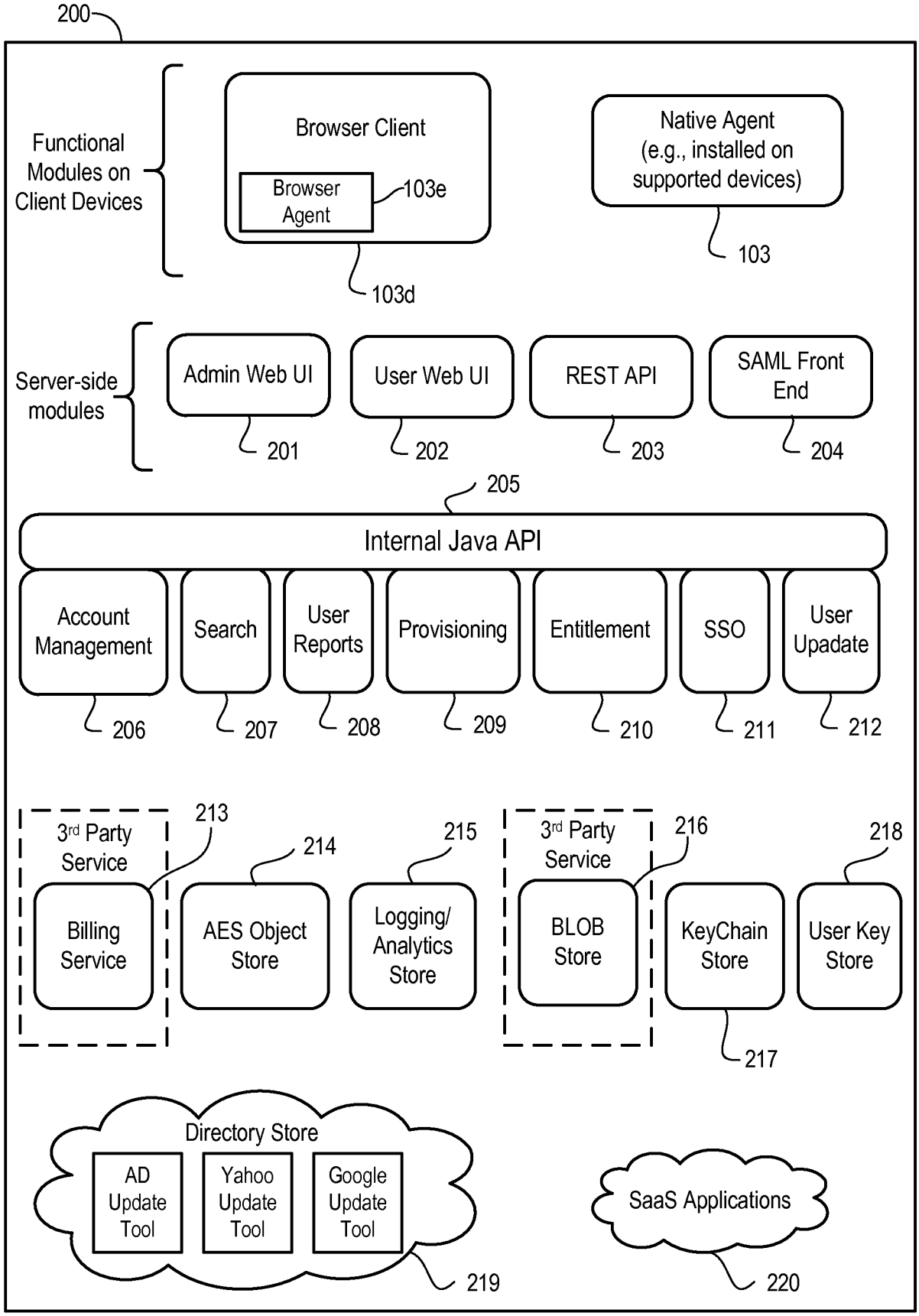
200
Functional Modules on Client Devices
Browser Client
Browser Agent
103e
103d
Native Agent (e.g., installed on supported devices)
103
Server-side modules
Admin Web UI
201
User Web UI
202
REST API
203
SAML Front End
204
205
Internal Java API
Account Management
206
Search
207
User Reports
208
Provisioning
209
Entitlement
210
SSO
211
User Upadate
212
3rd Party Service
213
Billing Service
214
AES Object Store
215
Logging/ Analytics Store
3rd Party Service
216
BLOB Store
KeyChain Store
217
218
User Key Store
Directory Store
AD Update Tool
Yahoo Update Tool
Google Update Tool
219
SaaS Applications
220

Fig. 2

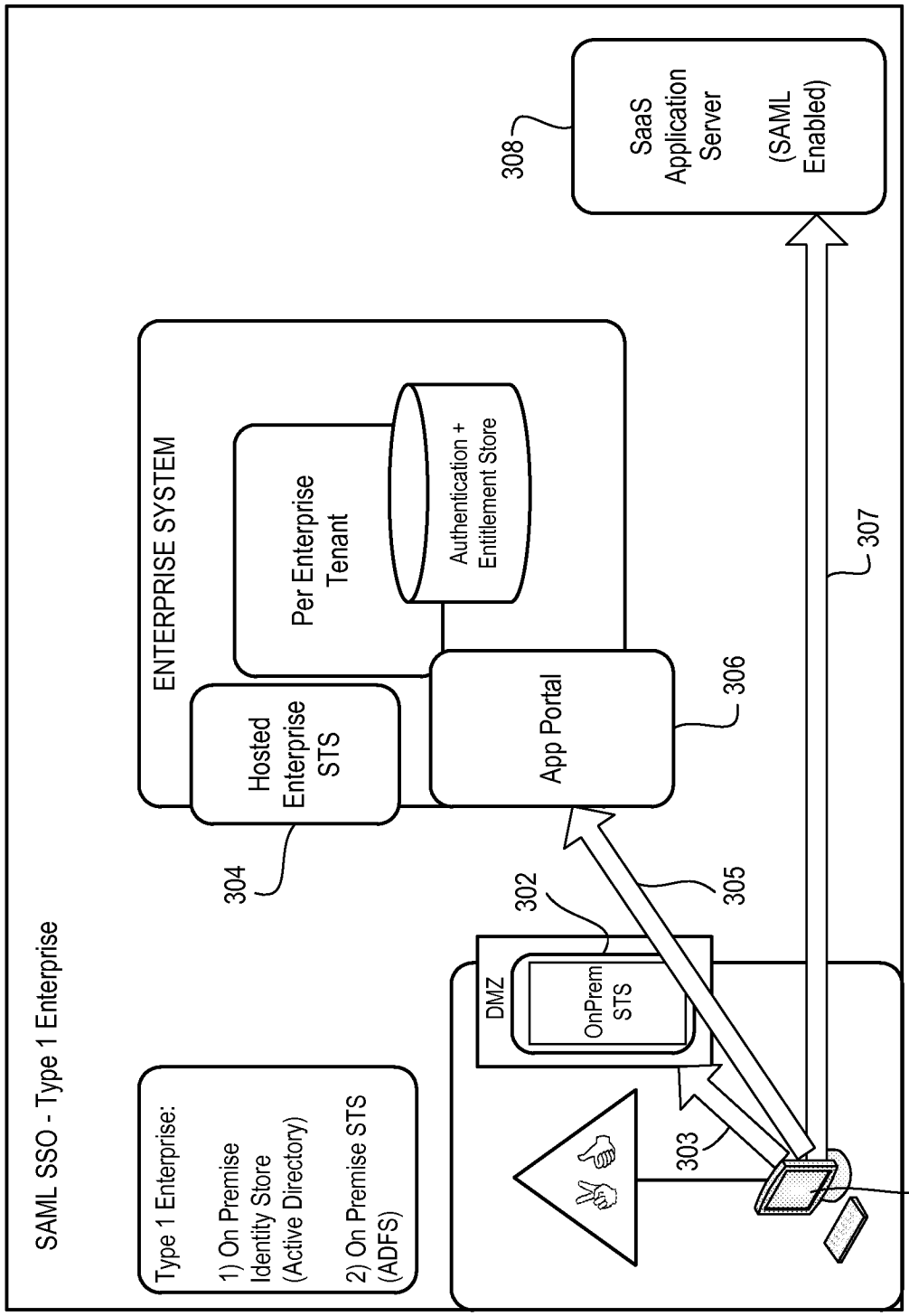
SAML SSO - Type 1 Enterprise
Type 1 Enterprise:
1) On Premise Identity Store (Active Directory)
2) On Premise STS (ADFS)
ENTERPRISE SYSTEM
Hosted Enterprise STS
Per Enterprise Tenant
Authentication + Entitlement Store
App Portal
DMZ
OnPrem STS
SaaS Application Server (SAML Enabled)
301
302
303
304
305
306
307
308

Fig. 3A

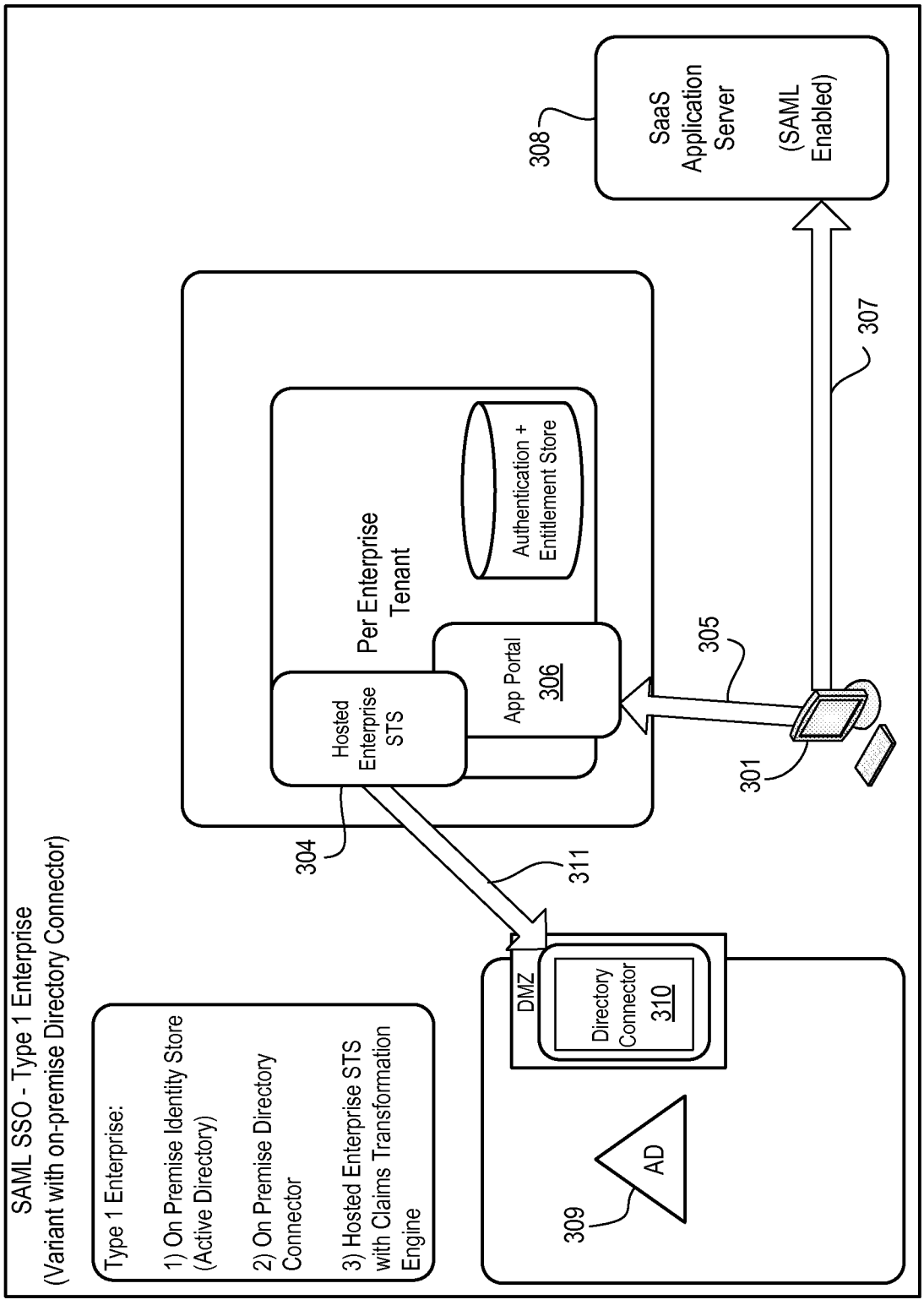
SAML SSO - Type 1 Enterprise
(Variant with on-premise Directory Connector)
Type 1 Enterprise:
1) On Premise Identity Store (Active Directory)
2) On Premise Directory Connector
3) Hosted Enterprise STS with Claims Transformation Engine
Per Enterprise Tenant
Hosted Enterprise STS
App Portal 306
Authentication + Entitlement Store
DMZ
Directory Connector 310
AD
309
304
311
305
301
307
308
SaaS Application Server (SAML Enabled)

Fig. 3B

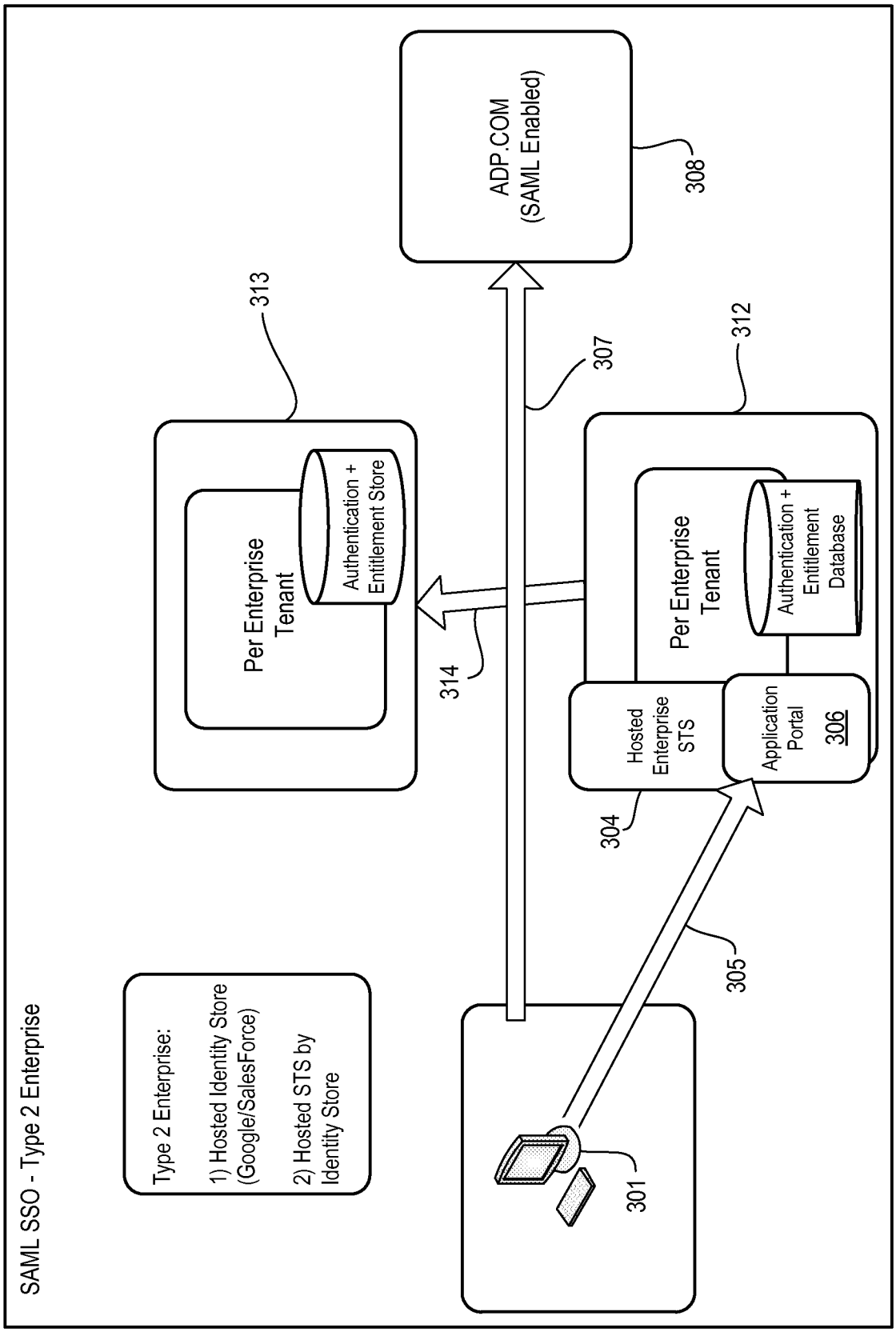
SAML SSO - Type 2 Enterprise
Type 2 Enterprise:
1) Hosted Identity Store (Google/SalesForce)
2) Hosted STS by Identity Store
ADP.COM (SAML Enabled)
308
313
Per Enterprise Tenant
Authentication + Entitlement Store
312
Per Enterprise Tenant
Authentication + Entitlement Database
Hosted Enterprise STS
Application Portal 306
307
314
304
305
301

Fig. 3C

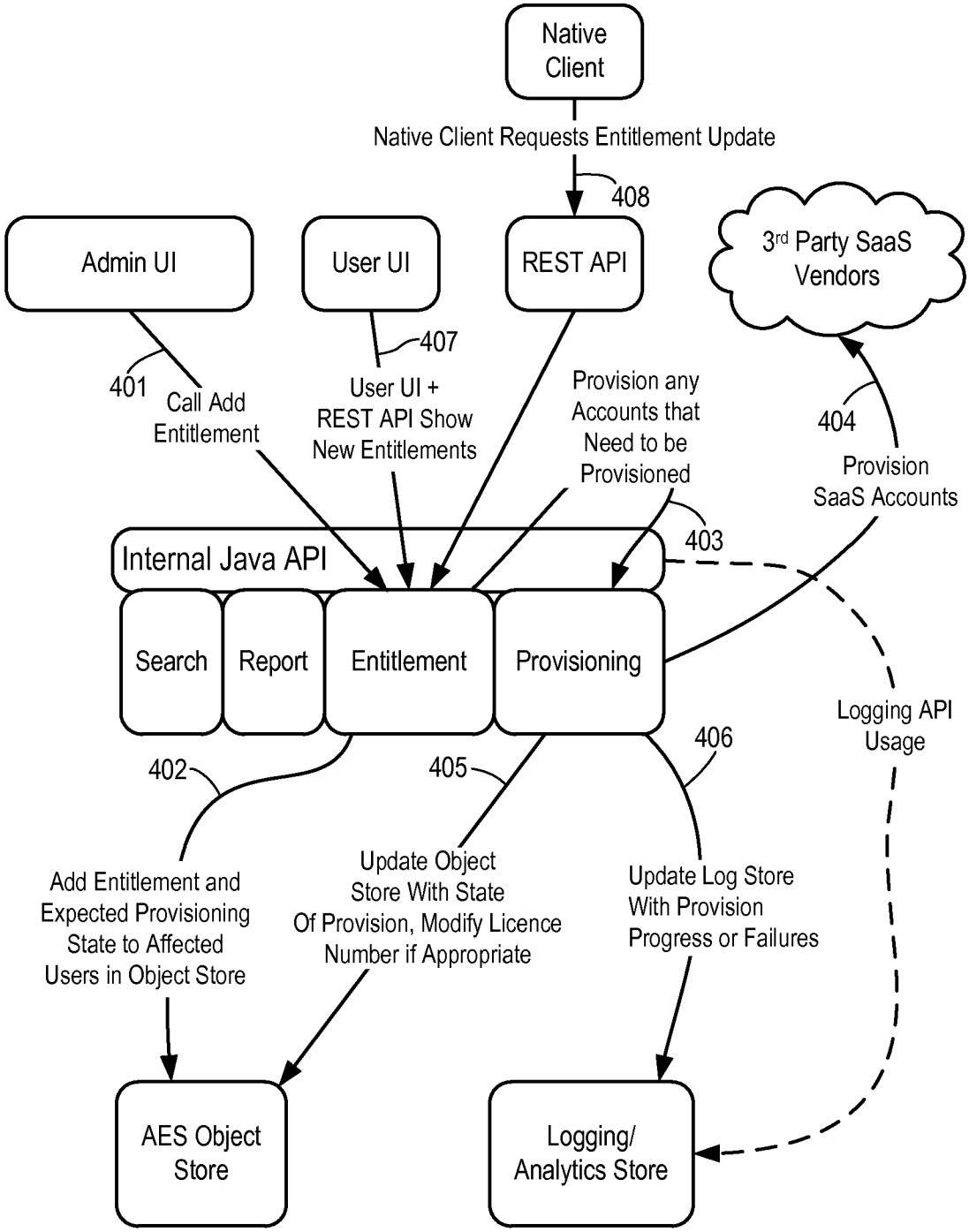
Native Client
Native Client Requests Entitlement Update
408
Admin UI
User UI
REST API
3rd Party SaaS Vendors
401
Call Add Entitlement
407
User UI + REST API Show New Entitlements
Provision any Accounts that Need to be Provisioned
404
Provision SaaS Accounts
403
Internal Java API
Search
Report
Entitlement
Provisioning
Logging API Usage
402
405
406
Add Entitlement and Expected Provisioning State to Affected Users in Object Store
Update Object Store With State Of Provision, Modify Licence Number if Appropriate
Update Log Store With Provision Progress or Failures
AES Object Store
Logging/ Analytics Store

Fig. 4

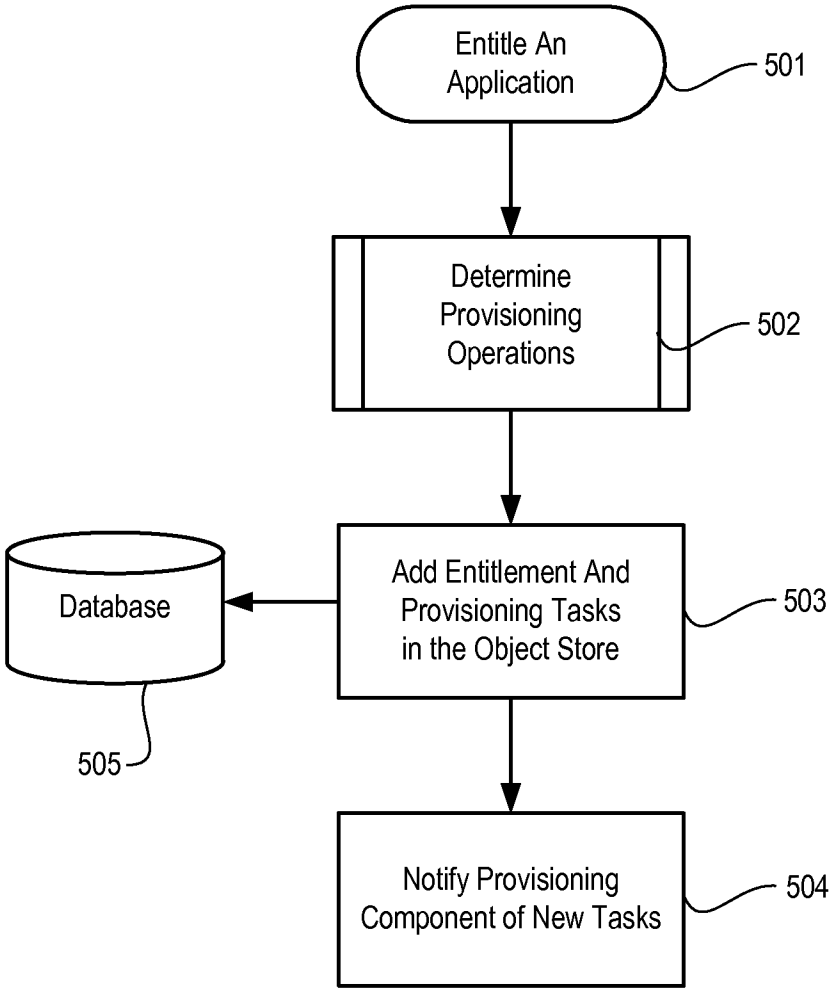
Entitle An Application
501
Determine Provisioning Operations
502
Add Entitlement And Provisioning Tasks in the Object Store
503
Database
505
Notify Provisioning Component of New Tasks
504

Fig. 5

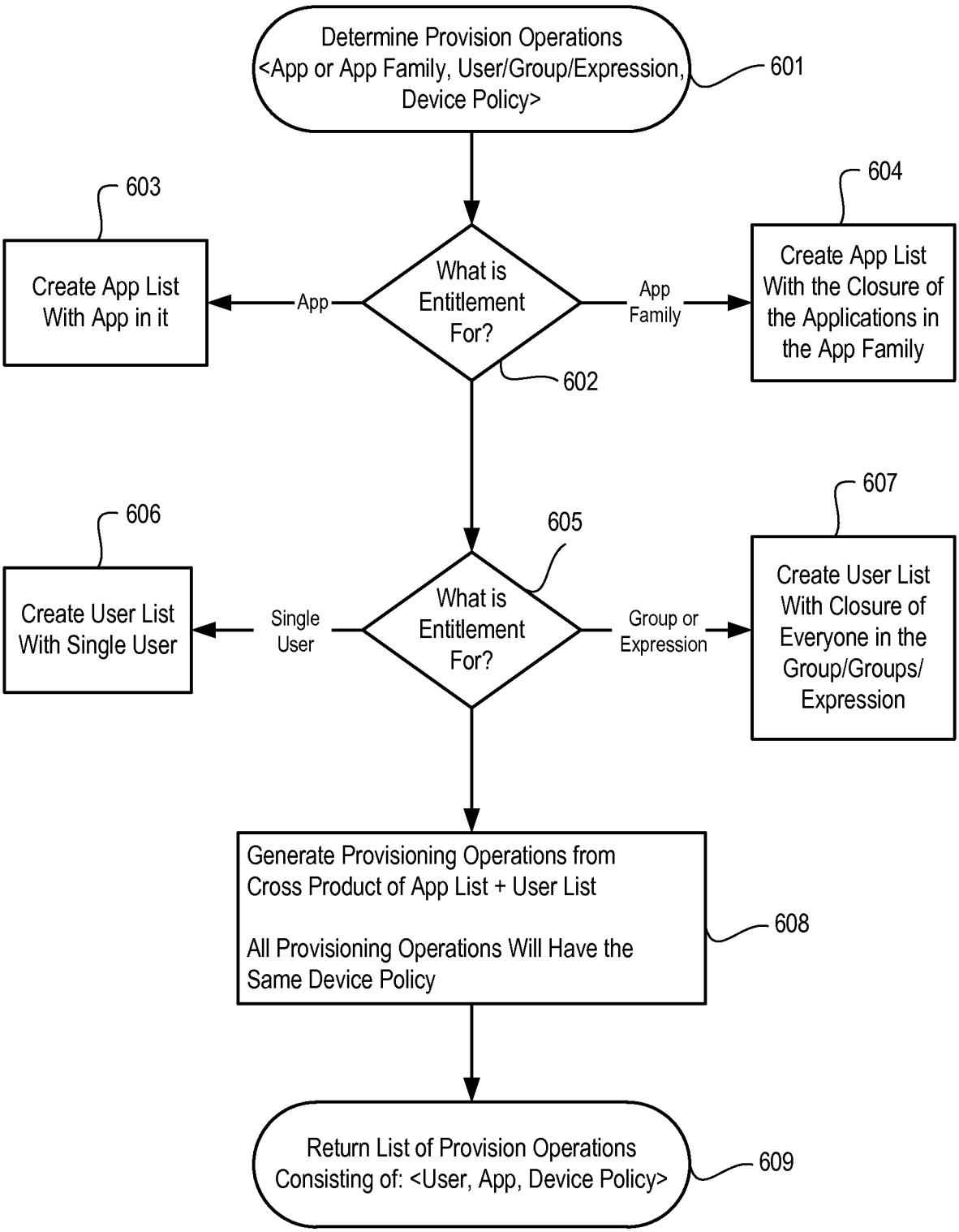
Determine Provision Operations <App or App Family, User/Group/Expression, Device Policy>
601
What is Entitlement For?
602
App
App Family
603
Create App List With App in it
604
Create App List With the Closure of the Applications in the App Family
What is Entitlement For?
605
Single User
Group or Expression
606
Create User List With Single User
607
Create User List With Closure of Everyone in the Group/Groups/ Expression
Generate Provisioning Operations from Cross Product of App List + User List
All Provisioning Operations Will Have the Same Device Policy
608
Return List of Provision Operations Consisting of: <User, App, Device Policy>
609

Fig. 6

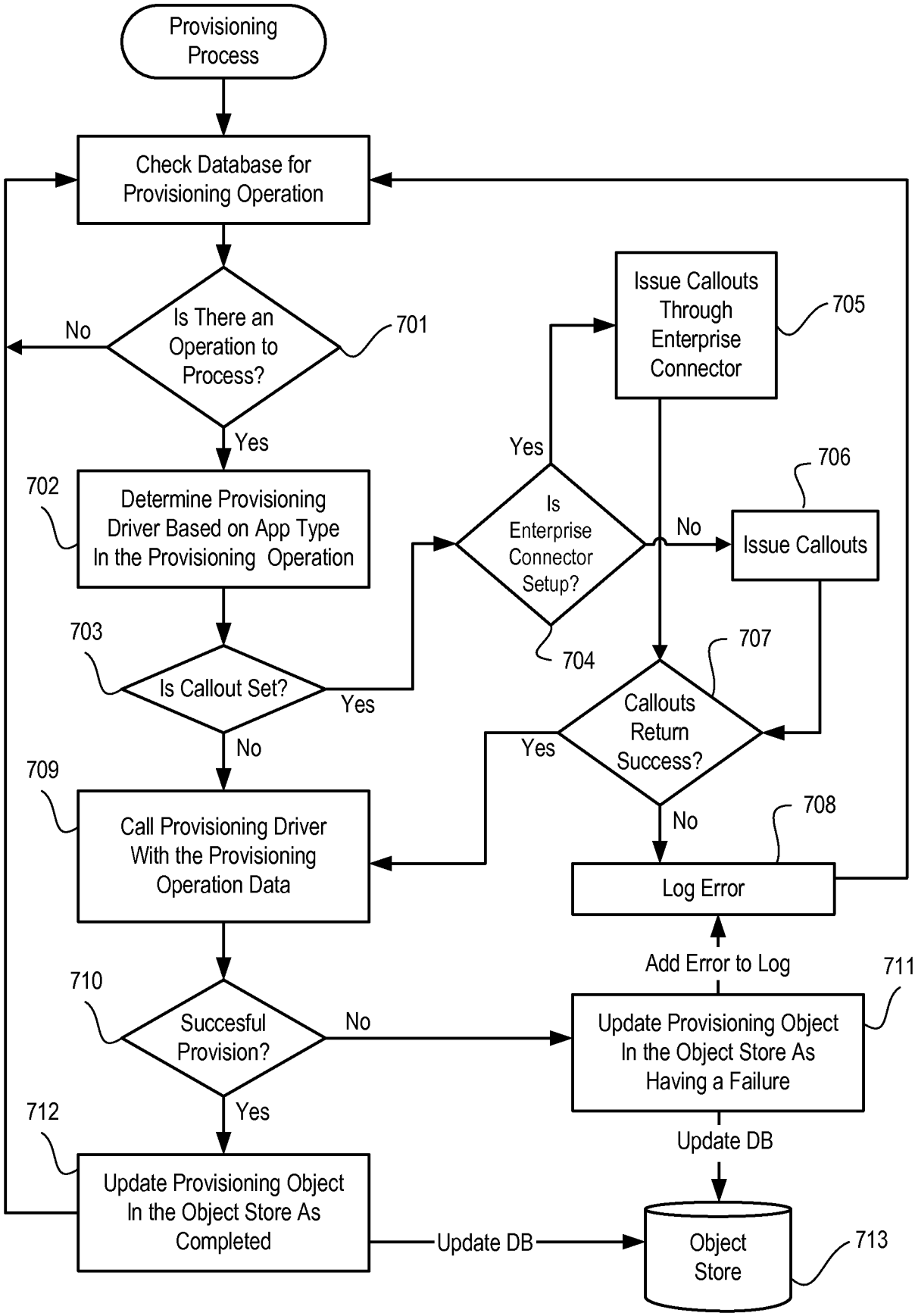
Provisioning Process
Check Database for Provisioning Operation
Is There an Operation to Process?
No
Yes
701
Determine Provisioning Driver Based on App Type In the Provisioning Operation
702
Is Callout Set?
703
Yes
No
Is Enterprise Connector Setup?
704
Yes
No
Issue Callouts Through Enterprise Connector
705
Issue Callouts
706
Callouts Return Success?
707
Yes
No
Call Provisioning Driver With the Provisioning Operation Data
709
Log Error
708
Add Error to Log
Succesful Provision?
710
No
Yes
Update Provisioning Object In the Object Store As Having a Failure
711
Update DB
Update Provisioning Object In the Object Store As Completed
712
Update DB
Object Store
713

Fig. 7

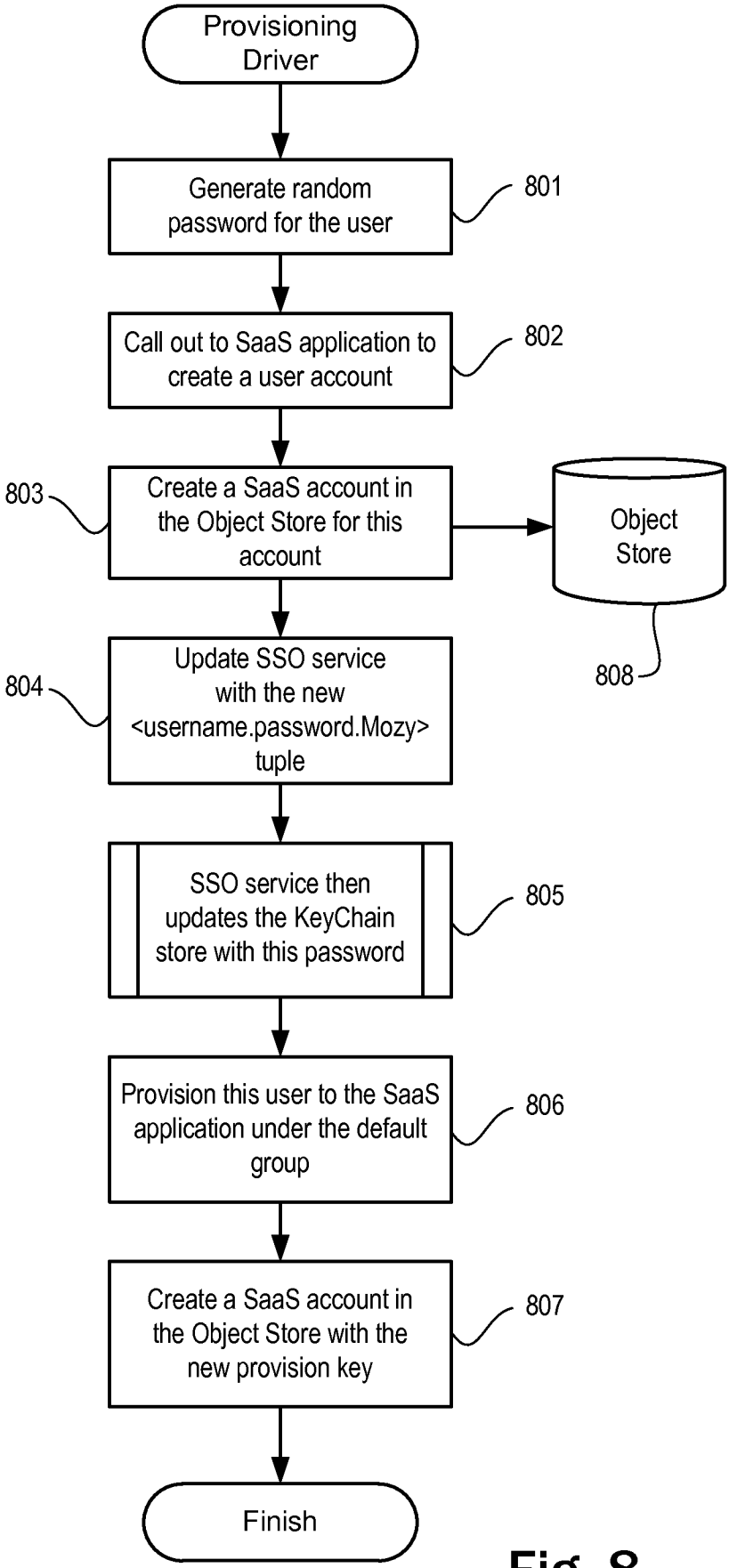
Provisioning Driver
Generate random password for the user
801
Call out to SaaS application to create a user account
802
803
Create a SaaS account in the Object Store for this account
Object Store
808
804
Update SSO service with the new <username.password.Mozy> tuple
SSO service then updates the KeyChain store with this password
805
Provision this user to the SaaS application under the default group
806
Create a SaaS account in the Object Store with the new provision key
807
Finish

Fig. 8

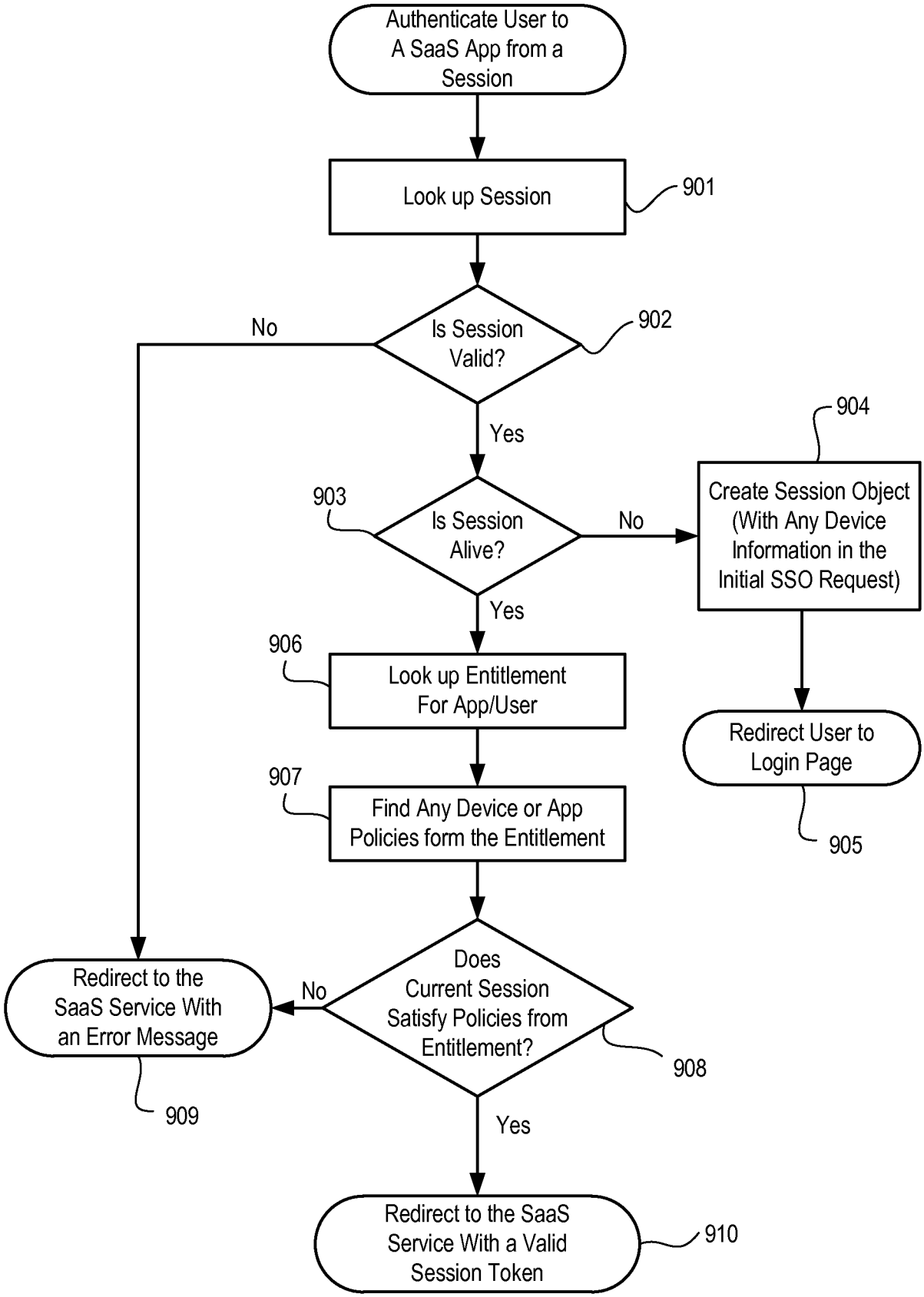
Authenticate User to A SaaS App from a Session
Look up Session
901
Is Session Valid?
902
No
Yes
903
Is Session Alive?
No
904
Create Session Object (With Any Device Information in the Initial SSO Request)
Redirect User to Login Page
905
Yes
906
Look up Entitlement For App/User
907
Find Any Device or App Policies form the Entitlement
Does Current Session Satisfy Policies from Entitlement?
908
No
Redirect to the SaaS Service With an Error Message
909
Yes
Redirect to the SaaS Service With a Valid Session Token
910

Fig. 9A

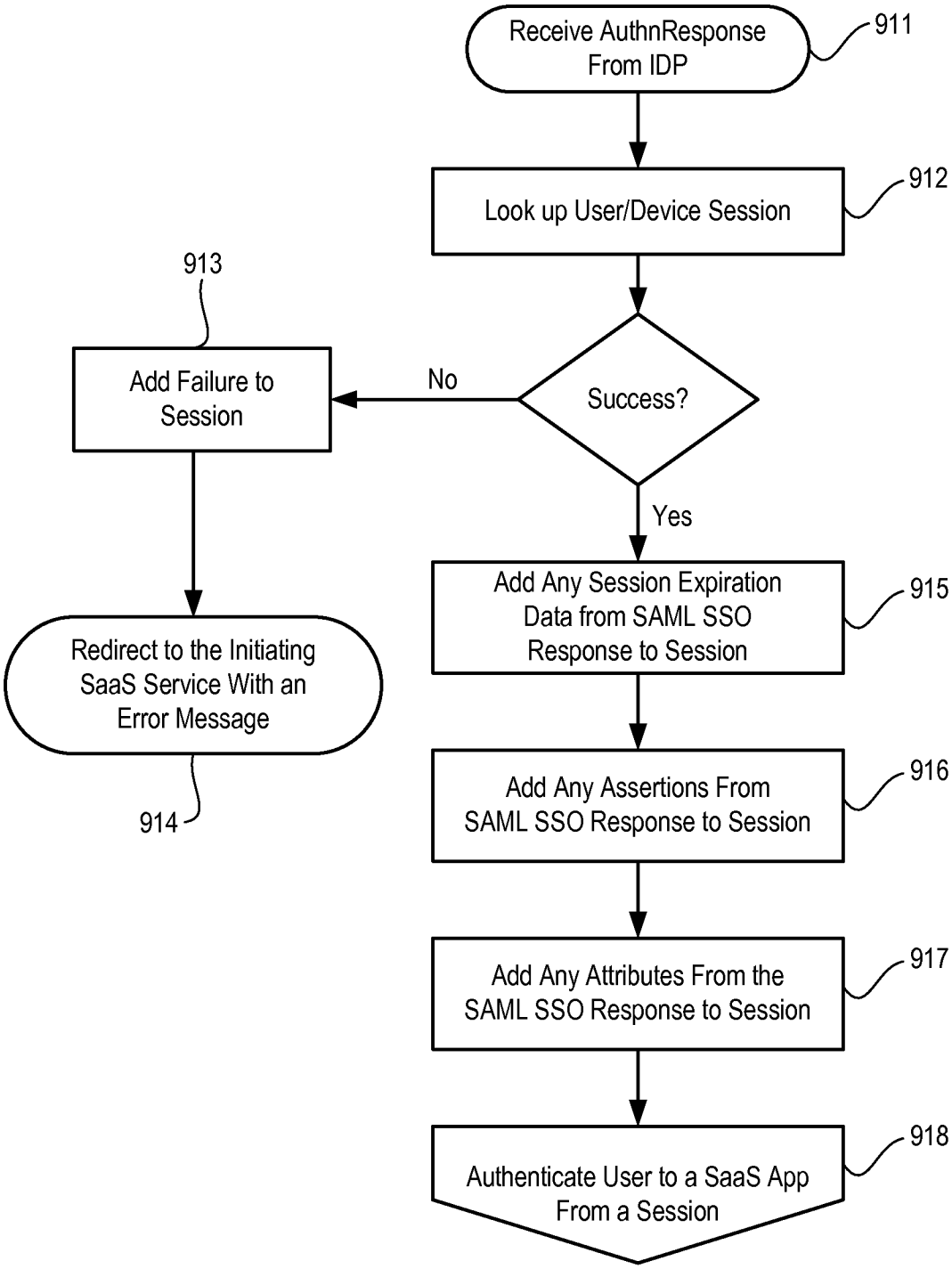
Receive AuthnResponse From IDP
911
Look up User/Device Session
912
Success?
No
Yes
913
Add Failure to Session
Redirect to the Initiating SaaS Service With an Error Message
914
Add Any Session Expiration Data from SAML SSO Response to Session
915
Add Any Assertions From SAML SSO Response to Session
916
Add Any Attributes From the SAML SSO Response to Session
917
Authenticate User to a SaaS App From a Session
918

Fig. 9B

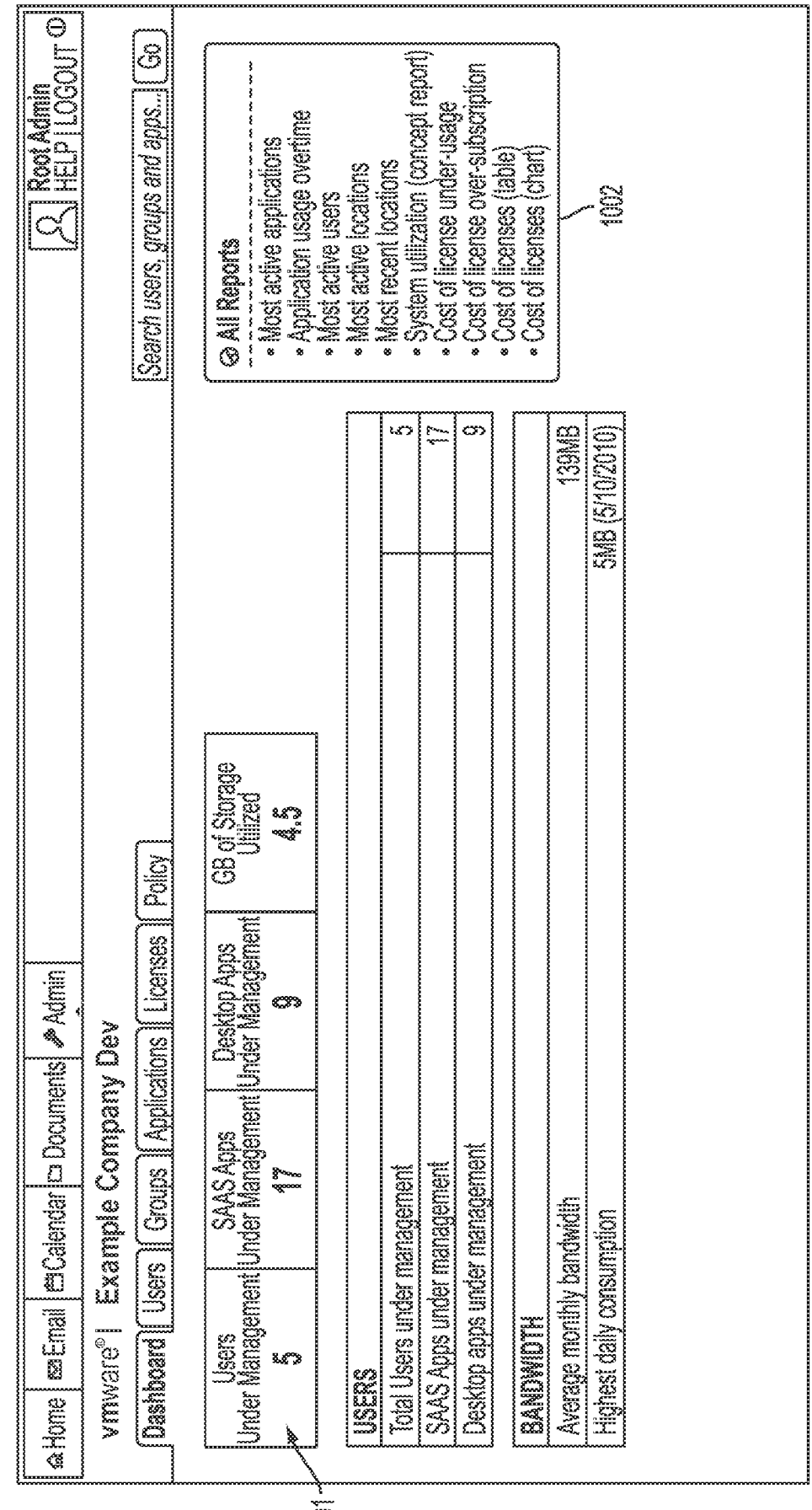
Home
Email
Calendar
Documents
Admin
Root Admin
HELP | LOGOUT
vmware | Example Company Dev
Dashboard
Users
Groups
Applications
Licenses
Policy
Search users, groups and apps...
Go
Users Under Management
5
SAAS Apps Under Management
17
Desktop Apps Under Management
9
GB of Storage Utilized
4.5
All Reports
• Most active applications
• Application usage overtime
• Most active users
• Most active locations
• Most recent locations
• System utilization (concept report)
• Cost of license under-usage
• Cost of license over-subscription
• Cost of licenses (table)
• Cost of licenses (chart)
1002
USERS
Total Users under management
5
SAAS Apps under management
17
Desktop apps under management
9
BANDWIDTH
Average monthly bandwidth
139MB
Highest daily consumption
5MB (5/10/2010)
1001

Fig. 10

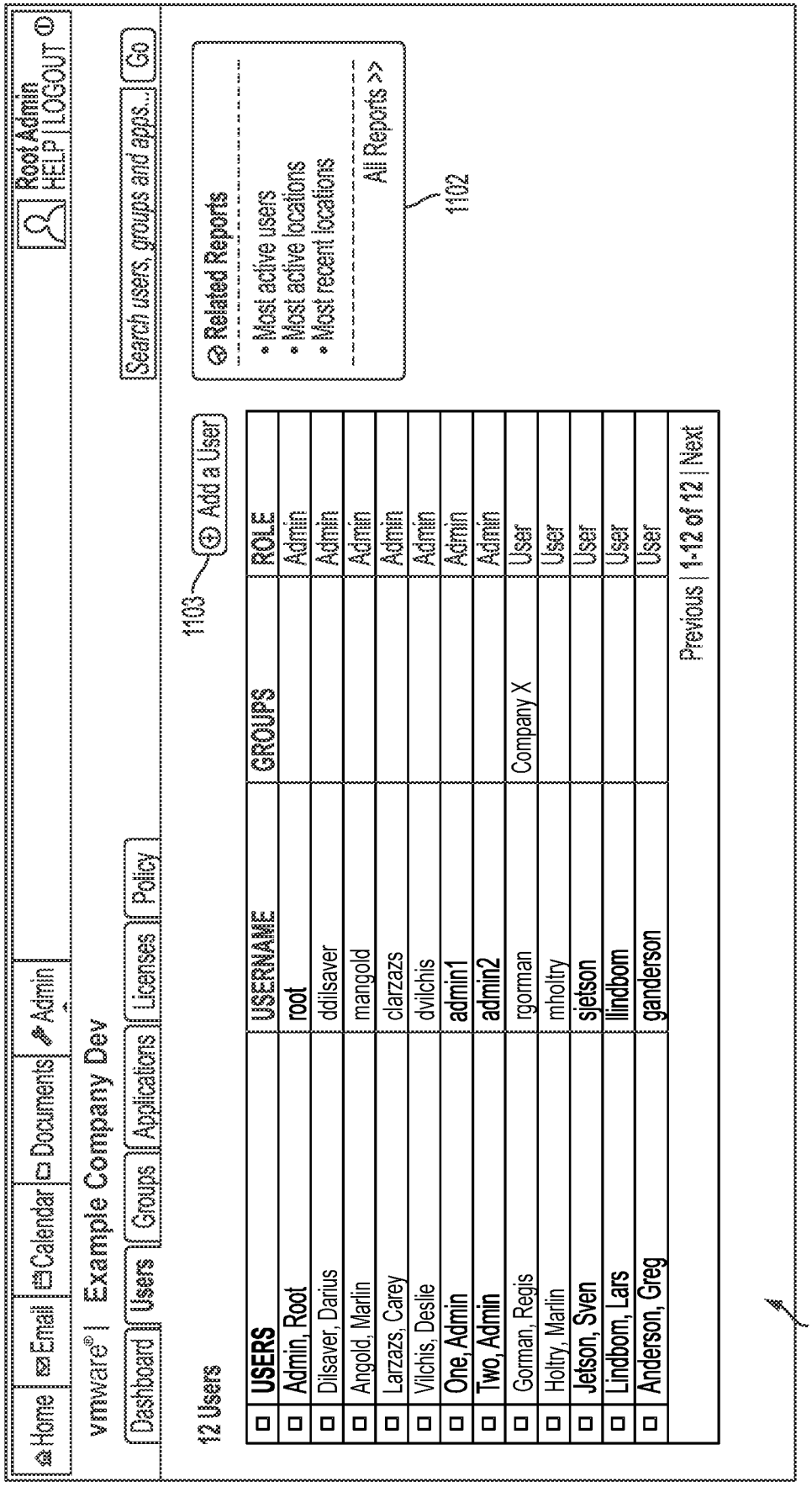
Home
Email
Calendar
Documents
Admin
Root Admin
HELP | LOGOUT
vmware | Example Company Dev
Dashboard
Users
Groups
Applications
Licenses
Policy
Search users, groups and apps...
Go
12 Users
Add a User
1103
Related Reports
• Most active users
• Most active locations
• Most recent locations
All Reports >>
1102
USERS
USERNAME
GROUPS
ROLE
Admin, Root
root
Admin
Dilsaver, Darius
ddilsaver
Admin
Angold, Marlin
mangold
Admin
Larzazs, Carey
clarzazs
Admin
Vilchis, Deslie
dvilchis
Admin
One, Admin
admin1
Admin
Two, Admin
admin2
Admin
Gorman, Regis
rgorman
Company X
User
Holtry, Marlin
mholtry
User
Jetson, Sven
sjetson
User
Lindbom, Lars
llindbom
User
Anderson, Greg
ganderson
User
Previous | 1-12 of 12 | Next
1101

Fig. 11

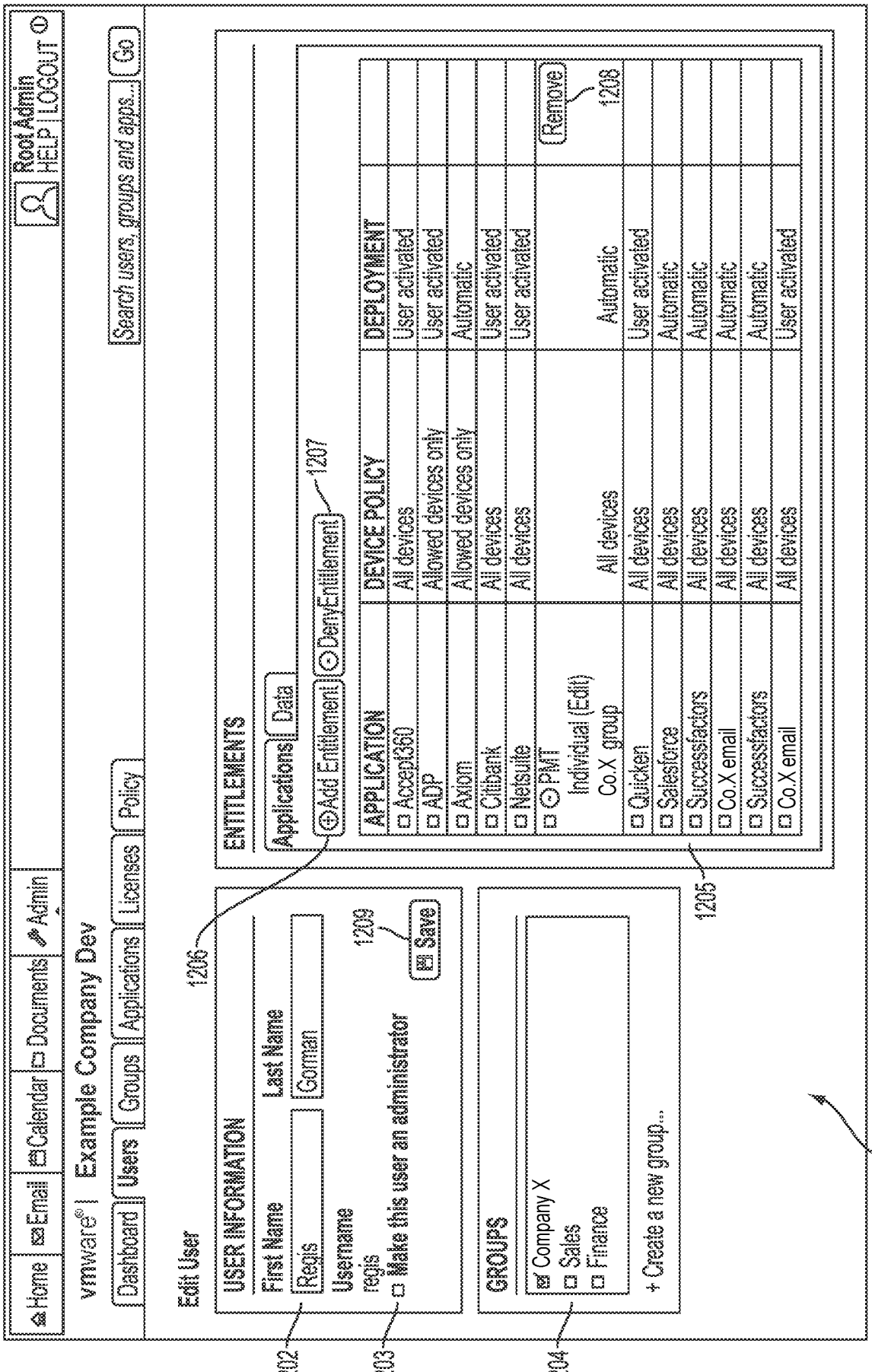
Home
Email
Calendar
Documents
Admin
Root Admin
HELP | LOGOUT
vmware® | Example Company Dev
Dashboard
Users
Groups
Applications
Licenses
Policy
Search users, groups and apps...
Go
Edit User
USER INFORMATION
First Name
Last Name
Regis
Gorman
Username
regis
Make this user an administrator
Save
1209
GROUPS
Company X
Sales
Finance
+ Create a new group...
1202
1203
1204
1206
ENTITLEMENTS
Applications
Data
Add Entitlement
Deny Entitlement
1207
APPLICATION
DEVICE POLICY
DEPLOYMENT
Accept360
All devices
User activated
ADP
Allowed devices only
User activated
Axiom
Allowed devices only
Automatic
Citibank
All devices
User activated
Netsuite
All devices
User activated
PMT
Individual (Edit)
Co.X group
All devices
Automatic
Remove
1208
Quicken
All devices
User activated
Salesforce
All devices
Automatic
Successfactors
All devices
Automatic
Co.X email
All devices
Automatic
Successfactors
All devices
Automatic
Co.X email
All devices
User activated
1205
1201

Fig. 12

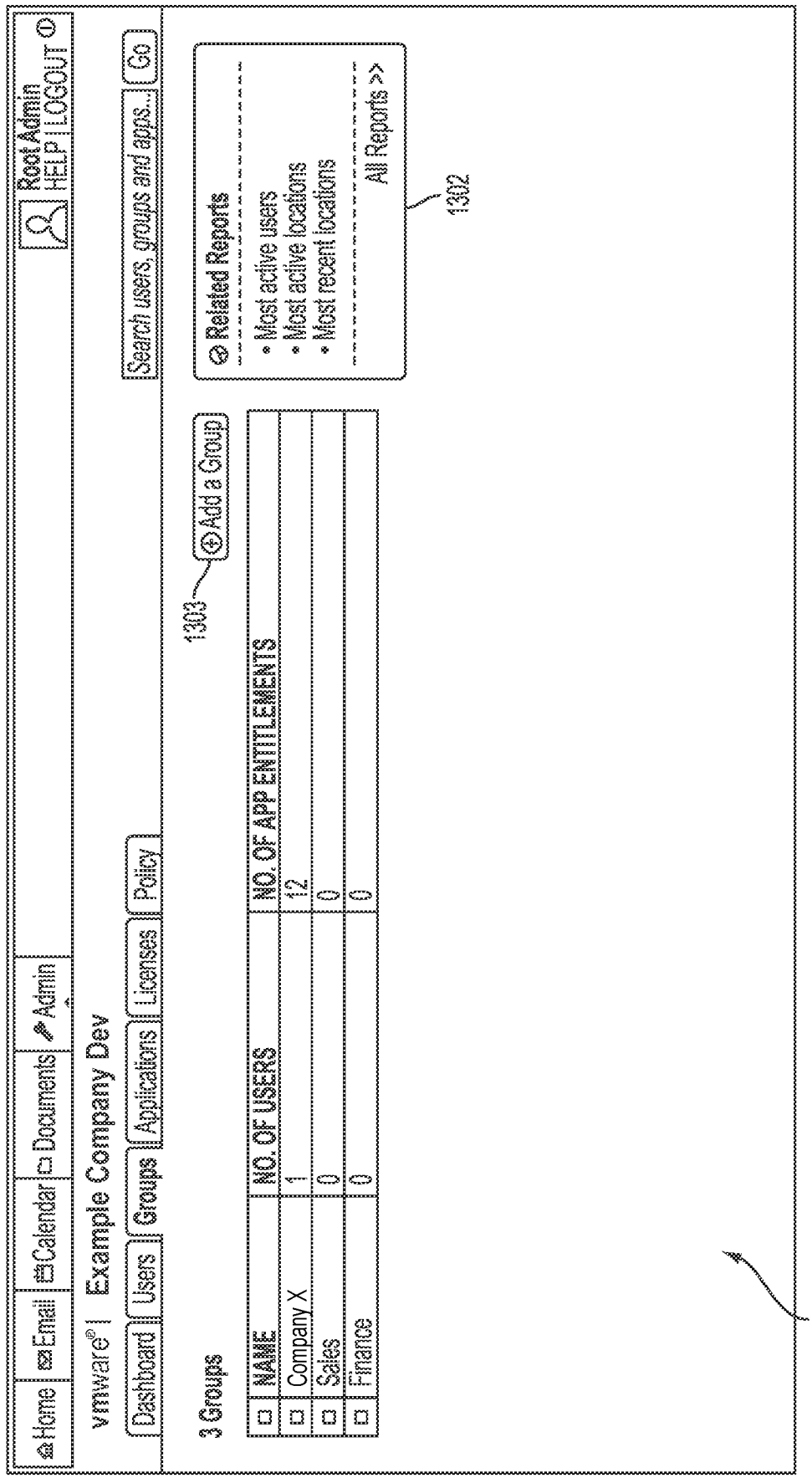
Home
Email
Calendar
Documents
Admin
Root Admin
HELP | LOGOUT
vmware® | Example Company Dev
Dashboard
Users
Groups
Applications
Licenses
Policy
Search users, groups and apps...
Go
3 Groups
Add a Group
Related Reports
• Most active users
• Most active locations
• Most recent locations
All Reports >>
NAME
NO. OF USERS
NO. OF APP ENTITLEMENTS
Company X
1
12
Sales
0
0
Finance
0
0
1301
1302
1303

Fig. 13

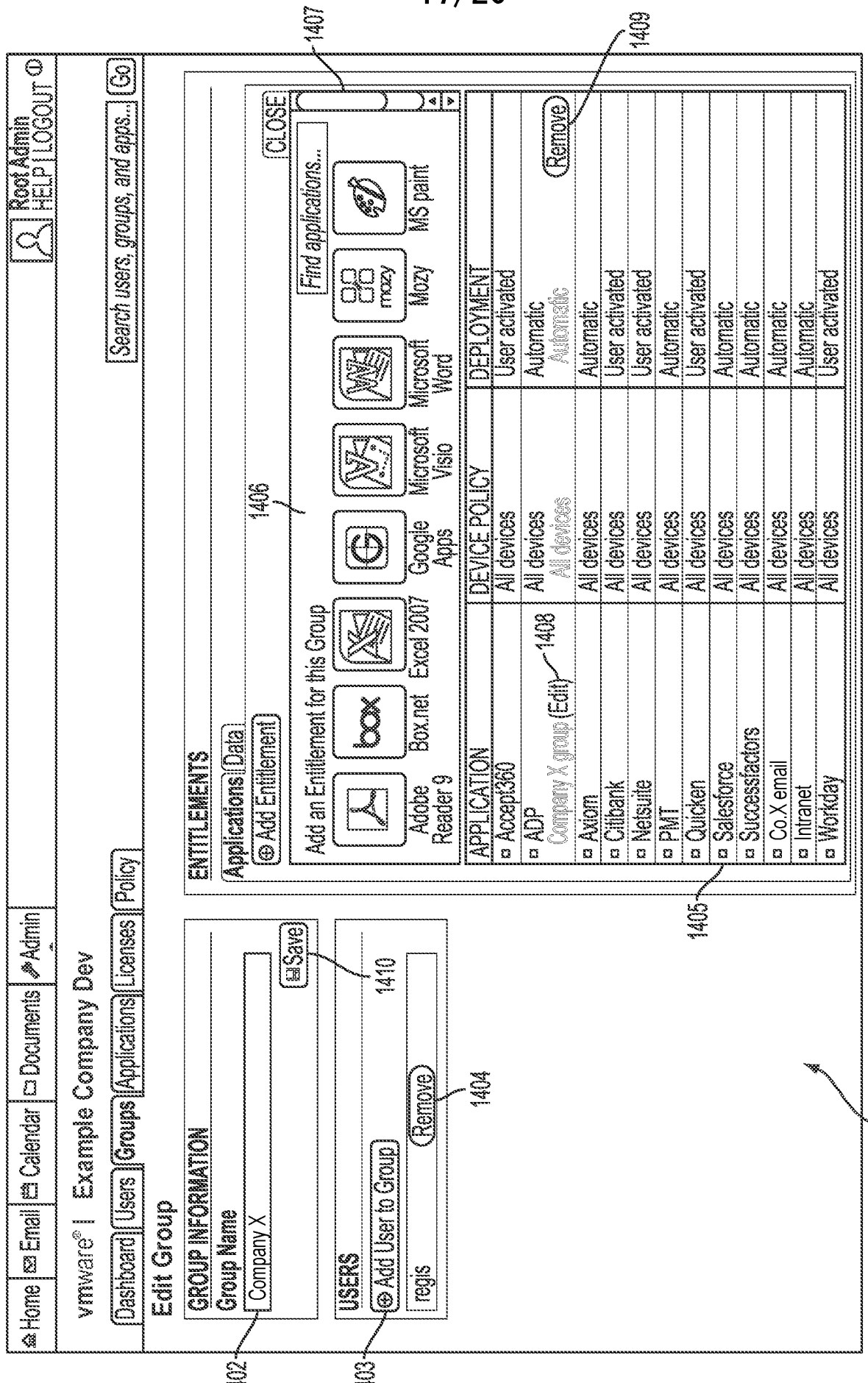
Home
Email
Calendar
Documents
Admin
Root Admin
HELP | LOGOUT
vmware® | Example Company Dev
Dashboard
Users
Groups
Applications
Licenses
Policy
Search users, groups, and apps...
Go
Edit Group
GROUP INFORMATION
Group Name
Company X
Save
USERS
Add User to Group
regis
Remove
ENTITLEMENTS
Applications
Data
Add Entitlement
Add an Entitlement for this Group
Adobe Reader 9
Box.net
Excel 2007
Google Apps
Microsoft Visio
Microsoft Word
Mozy
MS paint
Find applications...
CLOSE
APPLICATION
DEVICE POLICY
DEPLOYMENT
Accept360
All devices
User activated
ADP
Company X group (Edit)
All devices
Automatic
Automatic
Remove
Axiom
All devices
Automatic
Citibank
All devices
User activated
Netsuite
All devices
User activated
PMT
All devices
Automatic
Quicken
All devices
User activated
Salesforce
All devices
Automatic
Successfactors
All devices
Automatic
Co.X email
All devices
Automatic
Intranet
All devices
Automatic
Workday
All devices
User activated
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410

Fig. 14

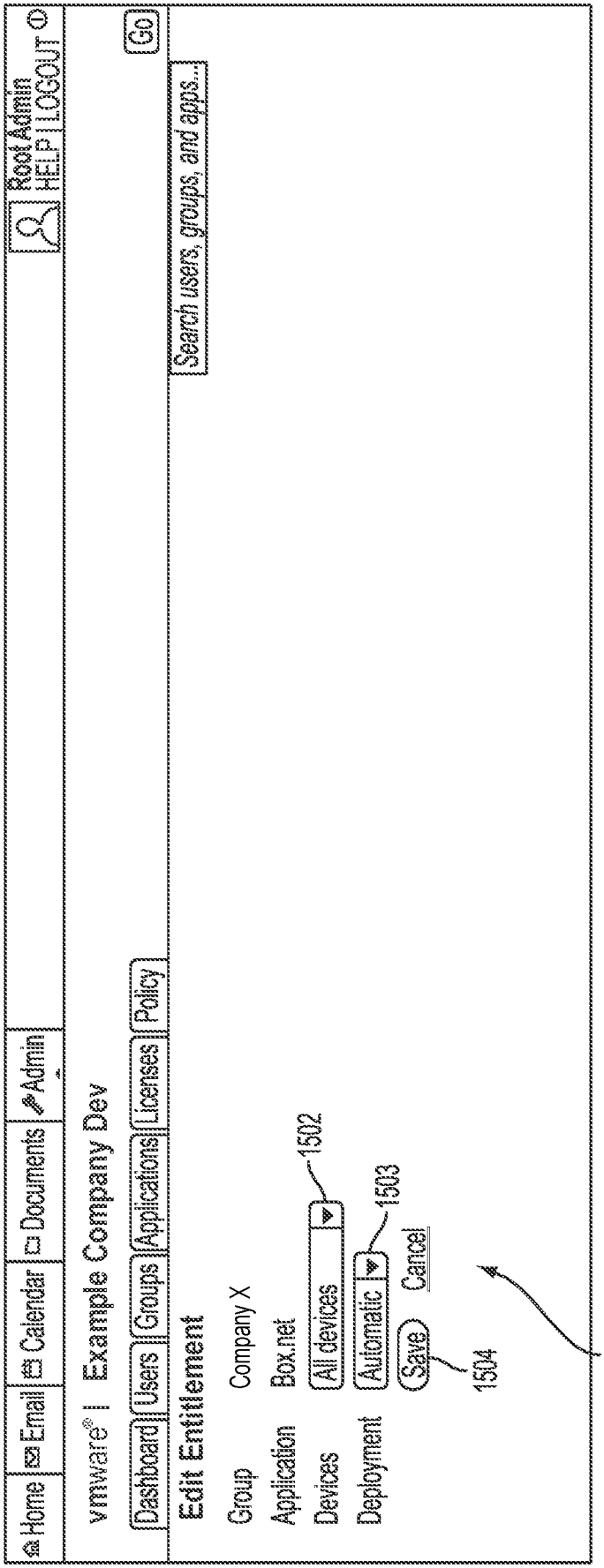
Home
Email
Calendar
Documents
Admin
Root Admin
HELP | LOGOUT
vmware | Example Company Dev
Dashboard
Users
Groups
Applications
Licenses
Policy
Search users, groups, and apps...
Go
Edit Entitlement
Group
Company X
Application
Box.net
Devices
All devices
Deployment
Automatic
Save
Cancel
1501
1502
1503
1504

Fig. 15

Home
Email
Calendar
Documents
Admin
Root Admin
HELP | LOGOUT
vmware® | Example Company Dev
Dashboard
Users
Groups
Applications
Licenses
Policy
Search users, groups, and apps...
Go
1603
1602
1604
1601
26 Applications
Add an Application
NAME/DESCRIPTION
TYPE
LICENSE REQUIRED
Accept360
Product requirements
Bookmark
1 of 100 total licenses being used
Adobe Reader 9
PDF reader (private)
RemoteApp
ADP
Employee Self Service
Bookmark
Axiom
Travel
Bookmark
License required
Box.net
Collaborative suite
Box.net
License required
Citibank
VMware credit card tracking
Bookmark
Excel 2007
Spreadsheet (private)
RemoteApp
Google Apps
Collaborative office software
Google Apps
0 of 2000 total licenses being used
Microsoft Visio
Diagramming (public)
RemoteApp
Microsoft Word
Create, edit, and access documents
Native
Mozy
Cloud Backup
Mozy
MS paint
pixel editor (public)
RemoteApp
Netsuite
Business management SaaS
Bookmark
3 of 50 total licenses being used
Photoshop
General editing
Native
Related Reports
Most active applications
Application usage over time
Cost of license under-usage
Cost of lices over subscription
Cost of licenses (table)
Cost of licenses (chart)
All Reports »
Help
If you want to add a license to an application, first make sure that "License required" is checked and then create the license.

Fig. 16

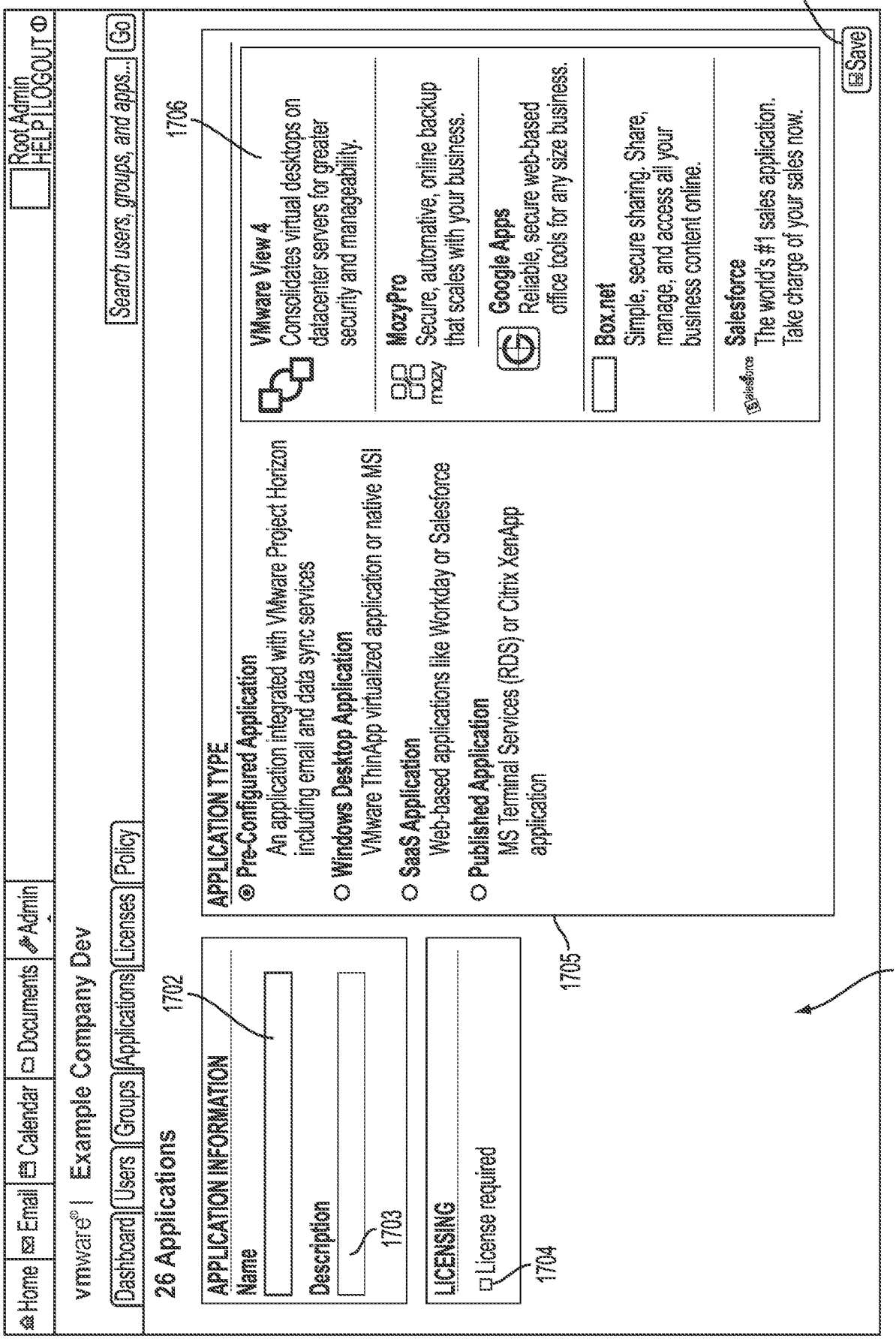
Home
Email
Calendar
Documents
Admin
Root Admin
HELP | LOGOUT
vmware® | Example Company Dev
Dashboard
Users
Groups
Applications
Licenses
Policy
Search users, groups, and apps...
Go
26 Applications
1702
1706
APPLICATION INFORMATION
Name
Description
1703
LICENSING
License required
1704
APPLICATION TYPE
Pre-Configured Application
An application integrated with VMware Project Horizon including email and data sync services
Windows Desktop Application
VMware ThinApp virtualized application or native MSI
SaaS Application
Web-based applications like Workday or Salesforce
Published Application
MS Terminal Services (RDS) or Citrix XenApp application
VMware View 4
Consolidates virtual desktops on datacenter servers for greater security and manageability.
MozyPro
Secure, automative, online backup that scales with your business.
Google Apps
Reliable, secure web-based office tools for any size business.
Box.net
Simple, secure sharing. Share, manage, and access all your business content online.
Salesforce
The world's #1 sales application. Take charge of your sales now.
Save
1707
1705
1701

Fig. 17

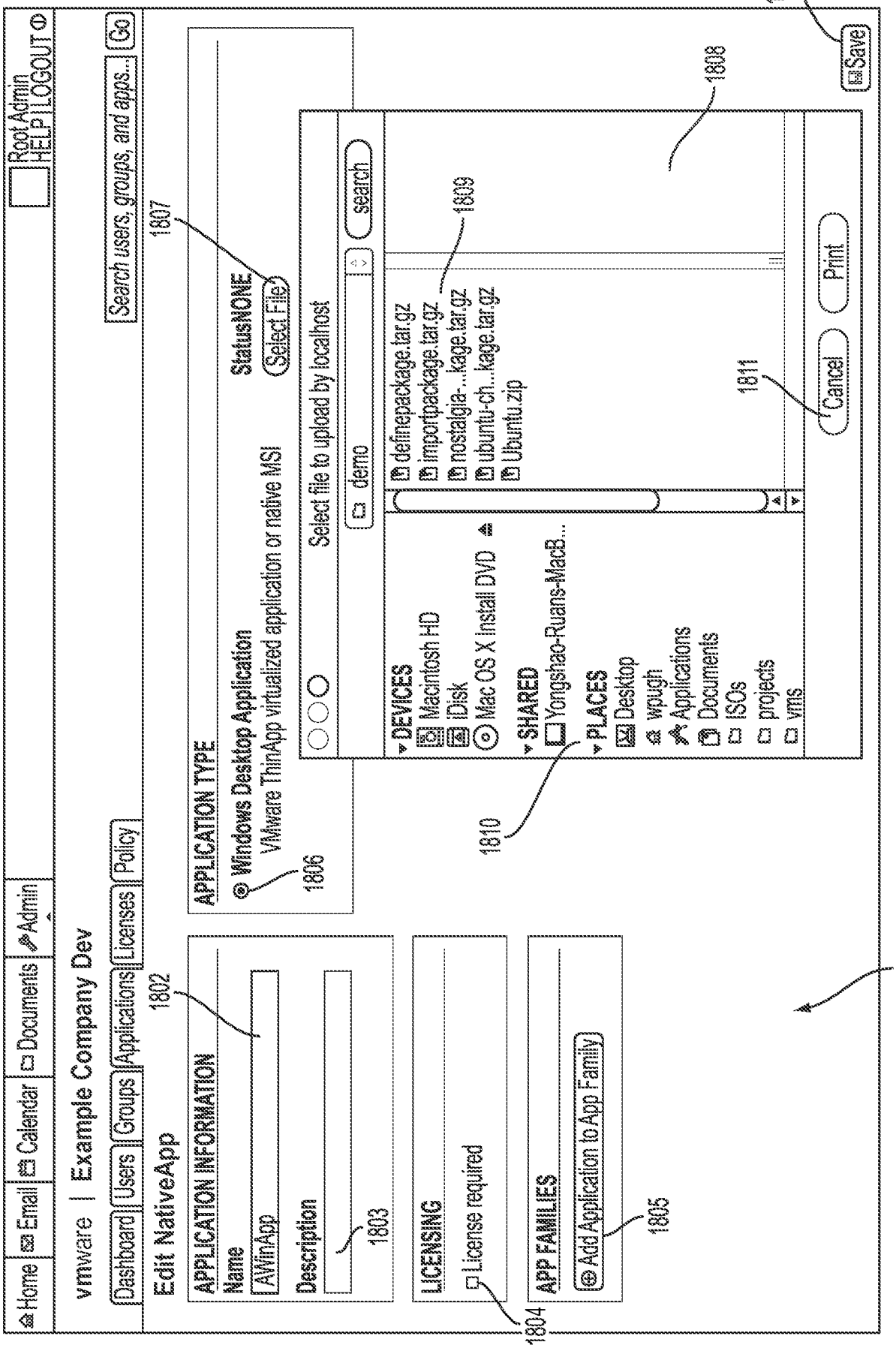
Home
Email
Calendar
Documents
Admin
Root Admin
HELP | LOGOUT
vmware | Example Company Dev
Dashboard
Users
Groups
Applications
Licenses
Policy
Search users, groups, and apps...
Go
Edit NativeApp
APPLICATION INFORMATION
Name
AWinApp
Description
LICENSING
License required
APP FAMILIES
Add Application to App Family
APPLICATION TYPE
Windows Desktop Application
VMware ThinApp virtualized application or native MSI
StatusNONE
Select File
Select file to upload by localhost
demo
search
DEVICES
Macintosh HD
iDisk
Mac OS X Install DVD
SHARED
Yongshao-Ruans-MacB...
PLACES
Desktop
wpugh
Applications
Documents
ISOs
projects
vms
definepackage.tar.gz
importpackage.tar.gz
nostalgia-...kage.tar.gz
ubuntu-ch...kage.tar.gz
Ubuntu.zip
Cancel
Print
Save
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812

Fig. 18

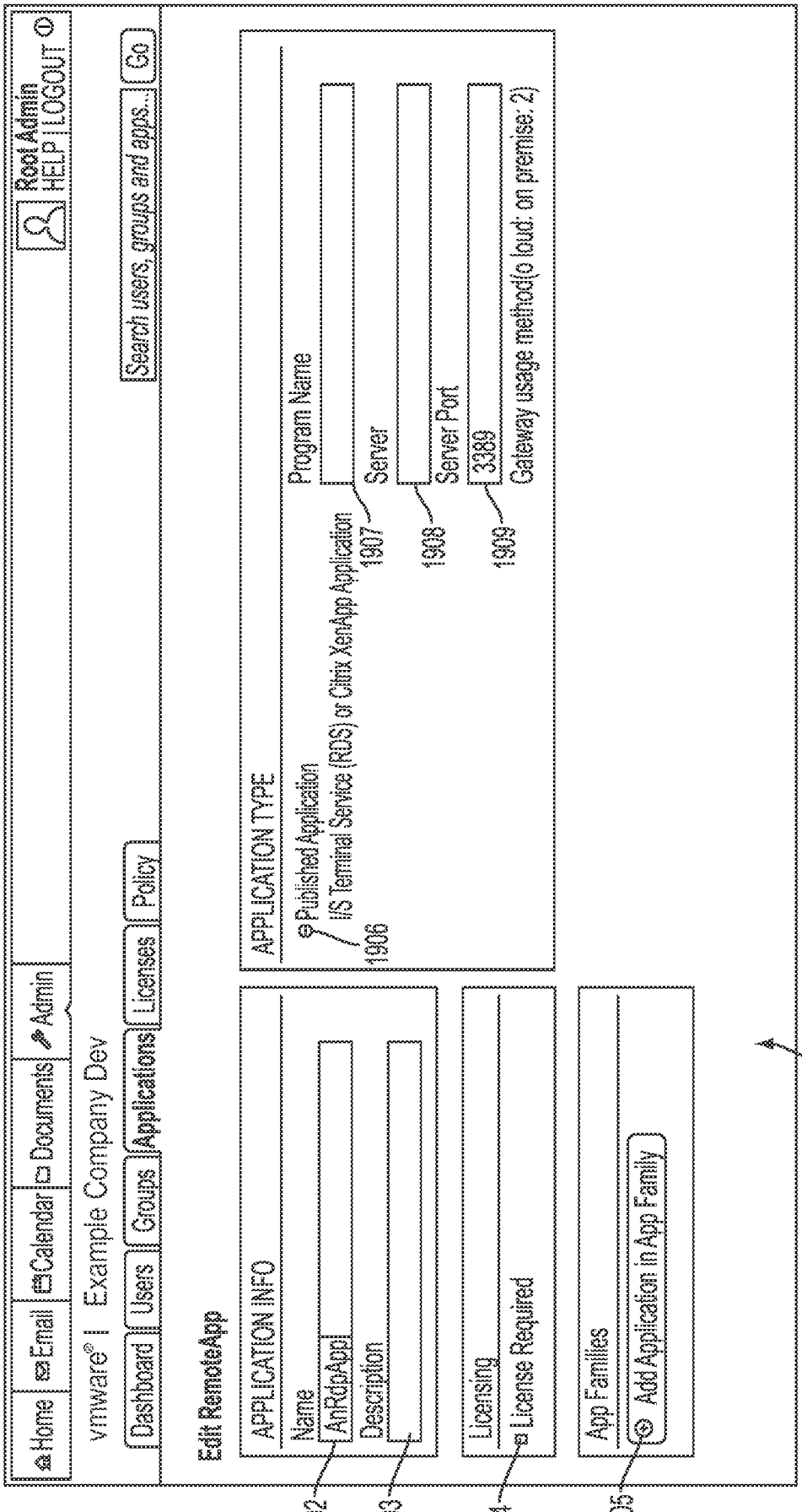
Home
Email
Calendar
Documents
Admin
Root Admin
HELP | LOGOUT
vmware® | Example Company Dev
Search users, groups and apps...
Go
Dashboard
Users
Groups
Applications
Licenses
Policy
Edit RemoteApp
APPLICATION INFO
Name
AnRdpAppl
Description
Licensing
License Required
App Families
Add Application in App Family
APPLICATION TYPE
Published Application
I/S Terminal Service (RDS) or Citrix XenApp Application
Program Name
Server
Server Port
3389
Gateway usage method(o (ucd: on premise: 2)
1901
1902
1903
1904
1905
1906
1907
1908
1909

Fig. 19

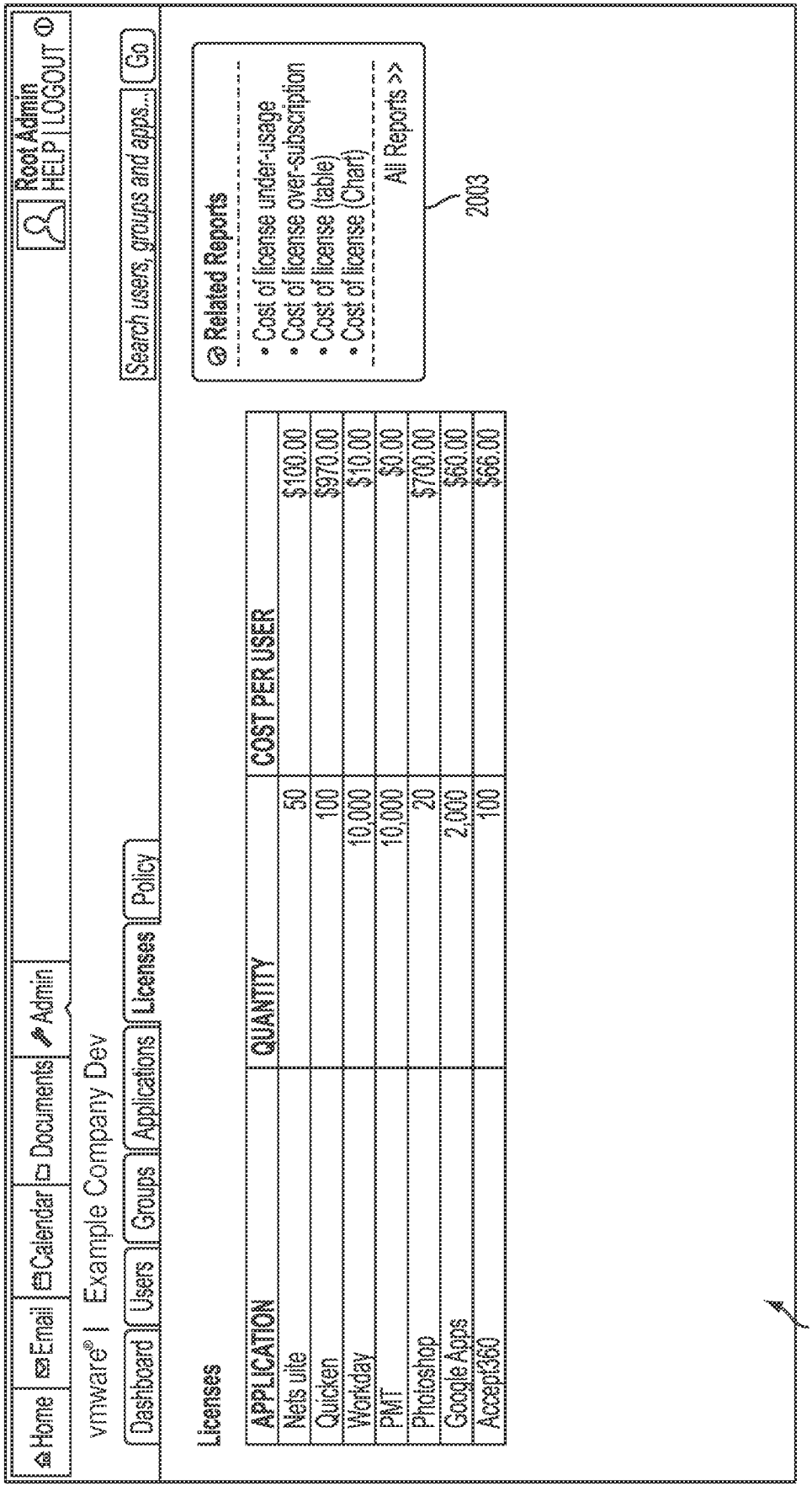
Home
Email
Calendar
Documents
Admin
Root Admin
HELP | LOGOUT
vmware® | Example Company Dev
Dashboard
Users
Groups
Applications
Licenses
Policy
Search users, groups and apps...
Go
Licenses
APPLICATION
QUANTITY
COST PER USER
Nets uite
50
$100.00
Quicken
100
$970.00
Workday
10,000
$10.00
PMT
10,000
$0.00
Photoshop
20
$700.00
Google Apps
2,000
$60.00
Accept360
100
$66.00
Related Reports
• Cost of license under-usage
• Cost of license over-subscription
• Cost of license (table)
• Cost of license (Chart)
All Reports >>
2003
2001

Fig. 20

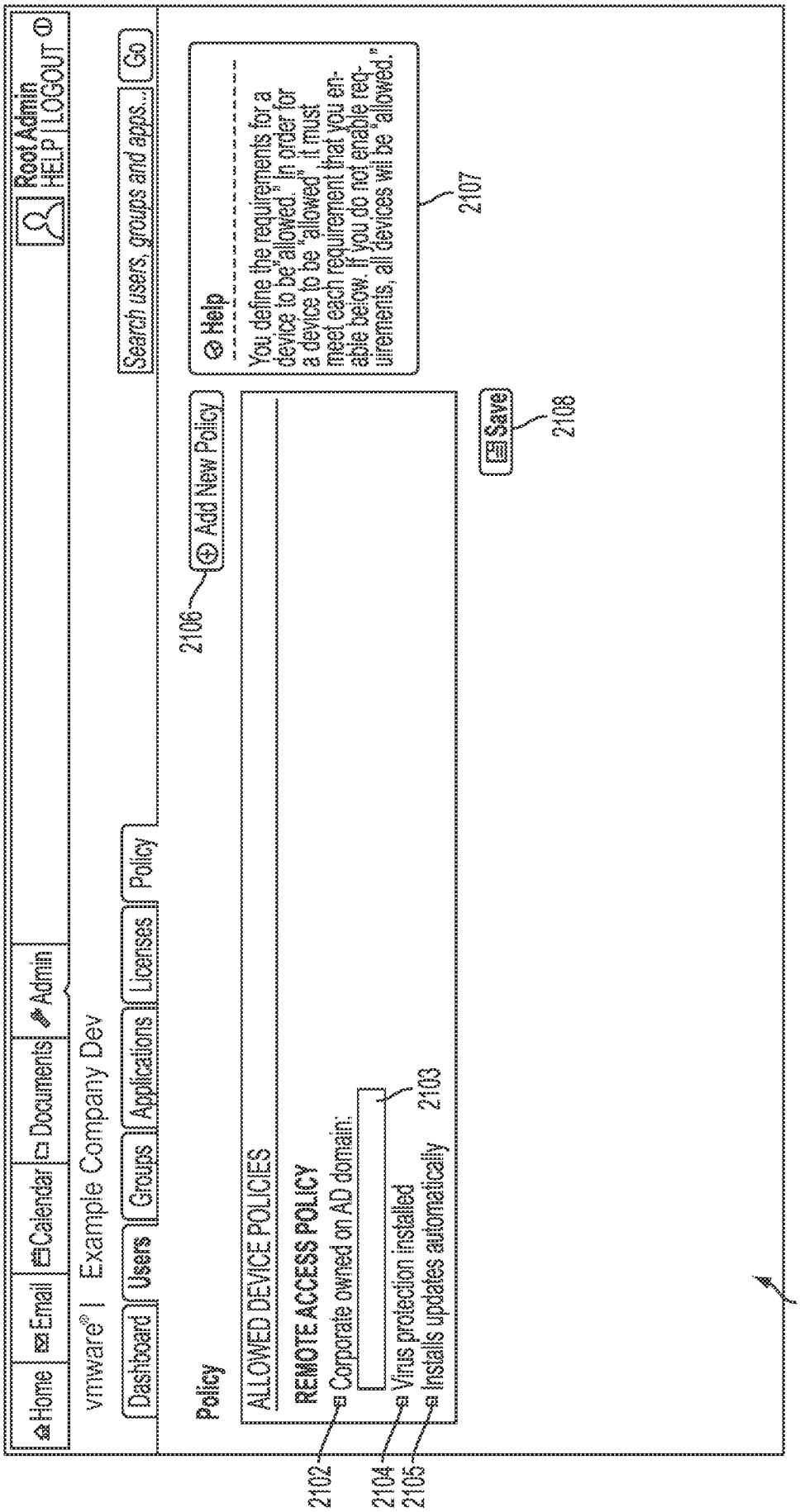
Home
Email
Calendar
Documents
Admin
Root Admin
HELP | LOGOUT
vmware® | Example Company Dev
Dashboard
Users
Groups
Applications
Licenses
Policy
Search users, groups and apps...
Go
Policy
Add New Policy
2106
Help
You define the requirements for a device to be "allowed." In order for a device to be "allowed", it must meet each requirement that you enable below. If you do not enable requirements, all devices will be "allowed."
2107
ALLOWED DEVICE POLICIES
REMOTE ACCESS POLICY
Corporate owned on AD domain
Virus protection installed
Installs updates automatically
2103
Save
2108
2102
2104
2105
2101

Fig. 21

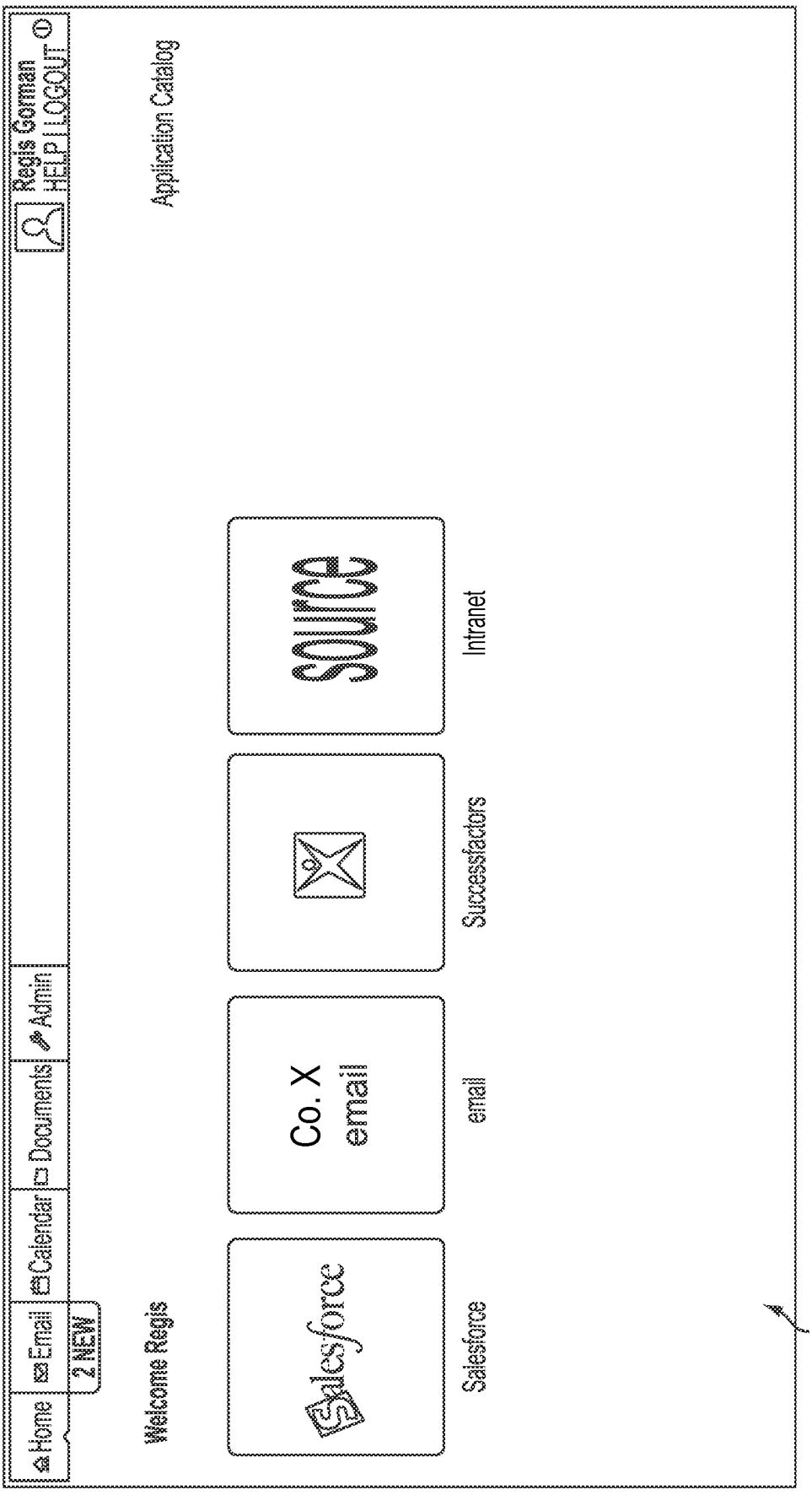
Regis Gorman
HELP | LOGOUT
Home
Email
2 NEW
Calendar
Documents
Admin
Application Catalog
Welcome Regis
Co. X email
Salesforce
email
Successfactors
Intranet
2201

Fig. 22

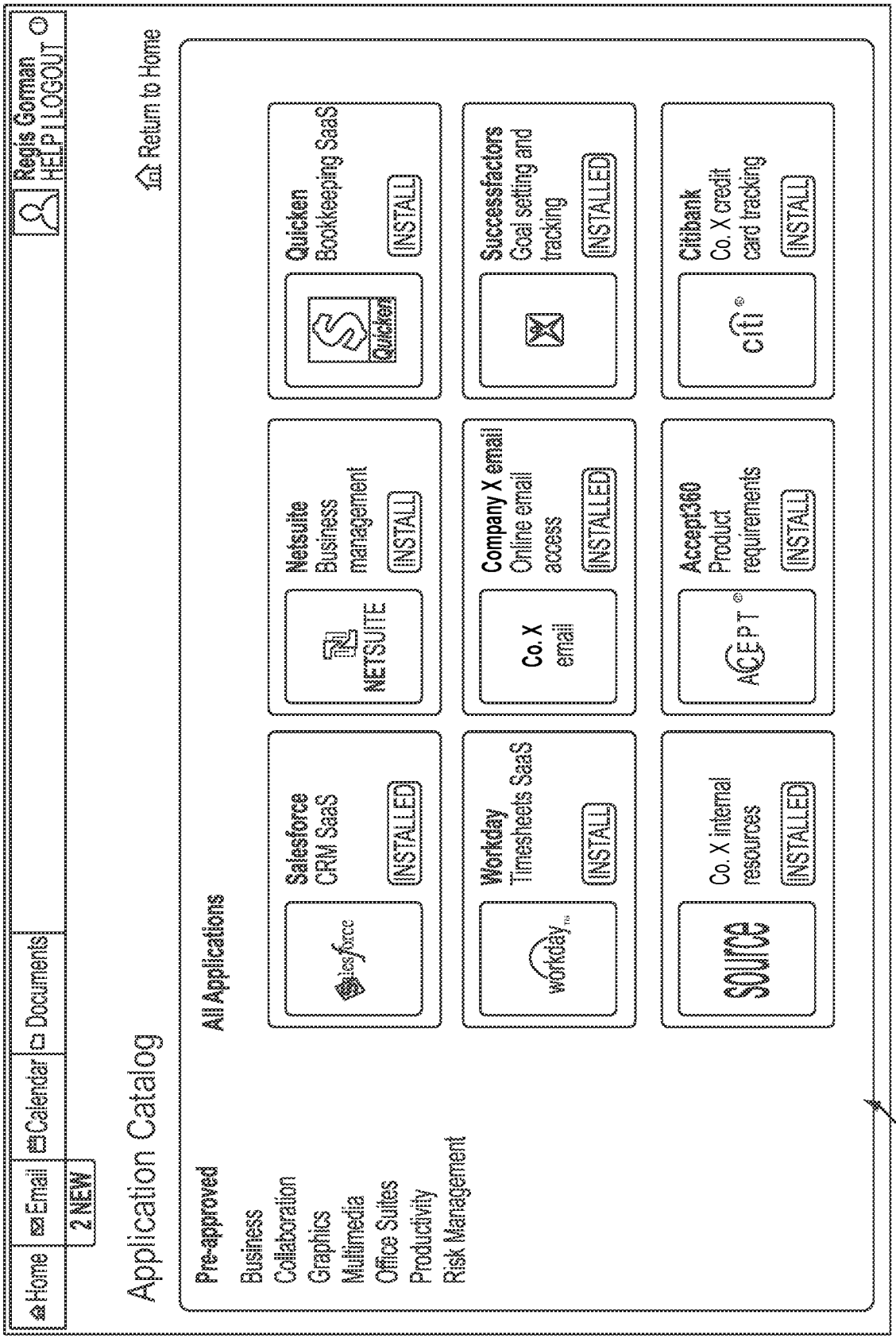
Home
Email
Calendar
Documents
Regis Corman
HELP LOGOUT
2 NEW
Application Catalog
Return to Home
Pre-approved
Business
Collaboration
Graphics
Multimedia
Office Suites
Productivity
Risk Management
All Applications
Salesforce
CRM SaaS
INSTALLED
Netsuite
Business management
INSTALL
Quicken
Bookkeeping SaaS
INSTALL
Workday
Timesheets SaaS
INSTALL
Company X email
Online email access
INSTALLED
Successfactors
Goal setting and tracking
INSTALLED
Co. X internal resources
INSTALLED
Accept360
Product requirements
INSTALL
Citibank
Co. X credit card tracking
INSTALL
2301

Fig. 23

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2011/049174

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/00
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

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 practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2008/147745 A1 (WILKINSON ANTHONY J [GB] ET AL) 19 June 2008 (2008-06-19) paragraph [0014]; figures 9a-9l paragraph [0066] paragraph [0069] paragraph [0090] paragraph [0278]	1-20
X	US 2007/239859 A1 (WILKINSON ANTHONY J [GB] ET AL) 11 October 2007 (2007-10-11) paragraphs [0086], [0090]; figures 1-3, 9a-9l	1-20
X	US 2002/138577 A1 (TENG JOAN C [US] ET AL) 26 September 2002 (2002-09-26) paragraph [0100]; figure 1	1-20

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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 document 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	Date of mailing of the international search report
21 November 2011	29/11/2011

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2011/049174

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	"Software as a Service", , 26 August 2010 (2010-08-26), XP55011009, Retrieved from the Internet: URL:http://en.wikipedia.org/w/index.php?title=Software_as_a_service&oldid=381046898 [retrieved on 2011-11-02] the whole document -----	1-20
A	Wikipedia: "Application streaming", , 15 July 2010 (2010-07-15), XP55011096, Retrieved from the Internet: URL:http://en.wikipedia.org/w/index.php?title=Application_streaming&oldid=373707909 [retrieved on 2011-11-03] the whole document -----	1-20
A	US 2008/288631 A1 (FAISAL ADIL [US] ET AL) 20 November 2008 (2008-11-20) paragraph [0027] -----	1-20

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2011/049174

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008147745 A1	19-06-2008	NONE	
US 2007239859 A1	11-10-2007	US 2007239859 A1	11-10-2007
		US 2008133777 A1	05-06-2008
US 2002138577 A1	26-09-2002	US 2002138577 A1	26-09-2002
		US 2011055673 A1	03-03-2011
US 2008288631 A1	20-11-2008	NONE	