



- (51) **International Patent Classification:**
G06F 21/44 (2013.01) *G06F 15/16* (2006.01)
- (21) **International Application Number:**
PCT/US2013/027258
- (22) **International Filing Date:**
22 February 2013 (22.02.2013)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
13/402,121 22 February 2012 (22.02.2012) US
- (71) **Applicant: MICROSOFT CORPORATION** [US/US];
One Microsoft Way, Redmond, WA 98052-6399 (US).
- (72) **Inventors: XUE, Gang;** c/o Microsoft Corporation, LCA -
International Patents, One Microsoft Way, Redmond, WA
98052-6399 (US). **SHI, Qiufang;** c/o Microsoft Corpora-
tion, LCA - International Patents, One Microsoft Way,
Redmond, WA 98052-6399 (US).

ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI,
NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU,
RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ,
TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA,
ZM, ZW.

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

Declarations under Rule 4.17:

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

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the
claims and to be republished in the event of receipt of
amendments (Rule 48.2(h))*

- (81) **Designated States** (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, 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,

(54) **Title:** VALIDATING LICENSE SERVERS IN VIRTUALIZED ENVIRONMENTS

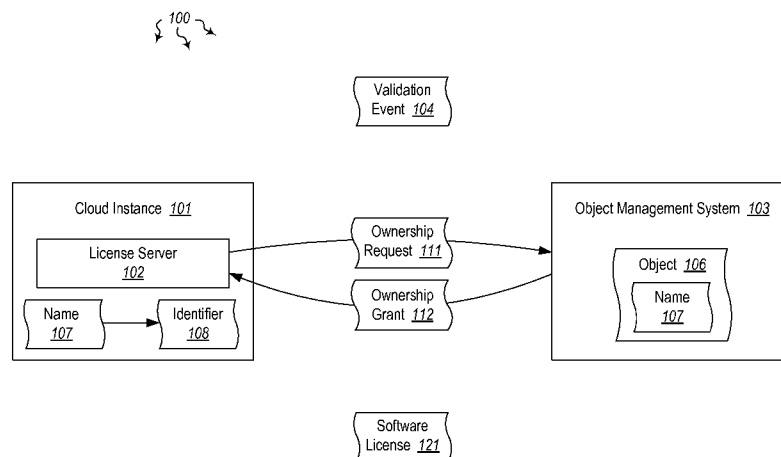


Figure 1

(57) **Abstract:** The present invention extends to methods, systems, and computer program products for validating license servers in virtualized environments. Embodiments of the invention leverage a set of features acquired or built in cloud computing environments to facilitate a software based solution providing uniqueness and immutability of a license server hosted in the cloud. Avoiding features of the underlying hardware systems results a much more flexible and reliable platform for hosting license servers. Features of a cloud storage service can be used to create a unique ID for a license server. Security and reliability of license servers hosted in a public cloud environment is also improved.

VALIDATING LICENSE SERVERS IN VIRTUALIZED ENVIRONMENTS**BACKGROUND****[0001] 1. Background and Relevant Art**

[0002] Computer systems and related technology affect many aspects of society.

5 Indeed, the computer system's ability to process information has transformed the way we live and work. Computer systems now commonly perform a host of tasks (e.g., word processing, scheduling, accounting, etc.) that prior to the advent of the computer system were performed manually. More recently, computer systems have been coupled to one another and to other electronic devices to form both wired and wireless computer networks
10 over which the computer systems and other electronic devices can transfer electronic data. Accordingly, the performance of many computing tasks is distributed across a number of different computer systems and/or a number of different computing environments.

[0003] In network environments, applications can be moved between computer systems. To deter piracy, many applications rely on a floating software license and license
15 servers. The floating software license is deployable on a license server so as to prevent from being abused. An application can be installed on any computer system. However, unless there is a valid license, the application will not run. To determine if an application can run on a computer system, the computer system queries the license server to determine if there is a valid license for the application.

20 [0004] To insure only a specified target license server is used, the target license server needs to be identified through a unique ID (e.g., fingerprint). In some environments, the unique ID is formulated distinctive and immutable features of an underlying hardware system, such as, for example, a Media Access Control ("MAC") address of a Network Interface Card ("NIC"). In other embodiments, a combination of hard and software
25 characters of a target license server can be used to formulate a unique ID. The unique ID is provided to application instances so that target license server can be queried.

[0005] Unfortunately, virtual systems do not inherently possess characteristics that can be used to generate a unique ID and thus deter piracy. In a virtual system, hardware characteristics, such as, for example, BIOS GUIDs and MAC addresses can be spoofed.
30 Spoofed hardware characteristics can then be used to clone a license server. Further, when operating in a cloud environment, cloud instances are transient virtual machines that are not bound to underlying hardware. Accordingly, hardware characteristics cannot be reliably used to formulate unique IDs in virtualized systems. Due at least in part to these

difficulties, many application End User License Agreements (“EULAs”) prohibit execution on virtual systems.

BRIEF SUMMARY

[0006] The present invention extends to methods, systems, and computer program products for validating license servers in virtualized environments. In some embodiments, a cloud instance hosts a license server in a cloud computing environment. The license server is validated as the target license server for a software license.

[0007] A validation event is detected for the license server. The validation event indicates that the license server is to check status as the target license server for the software license. The software license is installable at a license server having a unique identifier. The unique identifier is based on a unique name associated with a cloud-based object within the cloud computing environment.

[0008] The license server requests to assume exclusive ownership of the cloud-based object in response to the event. The cloud-based object has exclusivity of ownership such that a single license server can own the cloud-based object at any time. The license server is granted exclusive ownership of the cloud-based object. The license server uses the unique name to establish the unique identifier as the identifier for the license server.

[0009] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0010] Additional features and advantages of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by the practice of the invention. The features and advantages of the invention may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. These and other features of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to

be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

[0012] Figure 1 illustrates an example computer architecture that facilitates validating license servers in virtualized environments.

[0013] Figure 2 illustrates a flow chart of an example method for validating license servers in virtualized environments.

[0014] Figure 3 illustrates another example computer architecture that facilitates validating license servers in virtualized environments.

DETAILED DESCRIPTION

[0015] The present invention extends to methods, systems, and computer program products for validating license servers in virtualized environments. In some embodiments, a cloud instance hosts a license server in a cloud computing environment. The license server is validated as the target license server for a software license.

[0016] A validation event is detected for the license server. The validation event indicates that the license server is to check status as the target license server for the software license. The software license is installable at a license server having a unique identifier. The unique identifier is based on a unique name associated with a cloud-based object within the cloud computing environment.

[0017] The license server requests to assume exclusive ownership of the cloud-based object in response to the event. The cloud-based object has exclusivity of ownership such that a single license server can own the cloud-based object at any time. The license server is granted exclusive ownership of the cloud-based object. The license server uses the unique name to establish the unique identifier as the identifier for the license server.

[0018] Embodiments of the present invention may comprise or utilize a special purpose or general-purpose computer including computer hardware, such as, for example, one or more processors and system memory, as discussed in greater detail below. Embodiments within the scope of the present invention also include physical and other computer-readable media for carrying or storing computer-executable instructions and/or data structures. Such computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer system. Computer-readable media that store computer-executable instructions are computer storage media (devices). Computer-readable media that carry computer-executable instructions are transmission media. Thus, by way of example, and not limitation, embodiments of the invention can

comprise at least two distinctly different kinds of computer-readable media: computer storage media (devices) and transmission media.

[0019] Computer storage media (devices) includes RAM, ROM, EEPROM, CD-ROM, solid state drives (“SSDs”) (e.g., based on RAM), Flash memory, phase-change
5 memory (“PCM”), other types of memory, other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer.

[0020] A “network” is defined as one or more data links that enable the transport of
10 electronic data between computer systems and/or modules and/or other electronic devices. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of hardwired or wireless) to a computer, the computer properly views the connection as a transmission medium. Transmissions media can include a network and/or data links which can be used to carry
15 desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer. Combinations of the above should also be included within the scope of computer-readable media.

[0021] Further, upon reaching various computer system components, program code
20 means in the form of computer-executable instructions or data structures can be transferred automatically from transmission media to computer storage media (devices) (or vice versa). For example, computer-executable instructions or data structures received over a network or data link can be buffered in RAM within a network interface module (e.g., a “NIC”), and then eventually transferred to computer system RAM and/or to less
25 volatile computer storage media (devices) at a computer system. Thus, it should be understood that computer storage media (devices) can be included in computer system components that also (or even primarily) utilize transmission media.

[0022] Computer-executable instructions comprise, for example, instructions and data
30 which, when executed at a processor, cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. The computer executable instructions may be, for example, binaries, intermediate format instructions such as assembly language, or even source code. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the

appended claims is not necessarily limited to the described features or acts described above. Rather, the described features and acts are disclosed as example forms of implementing the claims.

[0023] Those skilled in the art will appreciate that the invention may be practiced in network computing environments with many types of computer system configurations, including, personal computers, desktop computers, laptop computers, message processors, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, mobile telephones, PDAs, tablets, pagers, routers, switches, and the like. The invention may also be practiced in distributed system environments where local and remote computer systems, which are linked (either by hardwired data links, wireless data links, or by a combination of hardwired and wireless data links) through a network, both perform tasks. In a distributed system environment, program modules may be located in both local and remote memory storage devices.

[0024] In this description and the following claims, “cloud computing” is defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. A cloud model can be composed of various characteristics (e.g., on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service, etc), service models (e.g., Software as a Service (“SaaS”), Platform as a Service (“PaaS”), Infrastructure as a Service (“IaaS”), and deployment models (e.g., private cloud, community cloud, public cloud, hybrid cloud, etc.).

[0025] Figure 1 illustrates an example computer architecture 100 that facilitates validating license servers in virtualized environments. Referring to Figure 1, computer architecture 100 includes cloud instance 101 and object management system 103. Cloud instance 101 can be one of number of cloud instances operating within a cloud. Cloud instance 101 and object management system 103 can be connected to one another over (or is part of) a network, such as, for example, a Local Area Network (“LAN”), a Wide Area Network (“WAN”), and even the Internet. Accordingly, cloud instance 101 and object management system 103 as well as any other connected computer systems and their components, can create message related data and exchange message related data (e.g., Internet Protocol (“IP”) datagrams and other higher layer protocols that utilize IP

datagrams, such as, Transmission Control Protocol (“TCP”), Hypertext Transfer Protocol (“HTTP”), Simple Mail Transfer Protocol (“SMTP”), etc.) over the network.

[0026] As depicted, cloud instance 101 includes license server 102. License server 102 can be a server that responds to license requests (e.g., from applications or other software modules). Object management system 103 can manage objects, such as, for example, accounts, files, users, etc, that are accessible from cloud instances (or from a cloud environment in general). Objects managed by object management system 103 can be given unique names. Objects managed by object management system 103 can also be configured for exclusive access (i.e., the objects can be owned by a single cloud instance at any time).

[0027] Software license 121 can be stored within computer architecture 100.

[0028] Figure 2 illustrates a flow chart of an example method 200 for validating license servers in virtualized environments. Method 200 will be described with respect to the components and data of computer architecture 100.

[0029] Method 200 includes an act of detecting a validation event for a license server, the validation event indicating that the license server is to check status as the target license server for the software license, the software license installable at a license server having a unique identifier, the unique identifier based on a unique name associated with a cloud-based object within the cloud computing environment (act 201). For example, validation event 104 can be detected. Validation event 104 can indicate that license server 102 is to check status as the target license server for software license 121. Software license 121 is installable at a license server having a unique identifier based on name 107 (the name of object 106). Object 106 can be an object with an exclusive access right. As such, object 106 (and corresponding properties) can be owned by a single license server at any time. Detecting a validation event can include detecting that software license 121 was stored in the cloud computing environment. Detecting a validation event can include detecting that a specified period of time has expired (e.g., since the status of license server was last checked).

[0030] Method 200 includes an act of license server requesting to assume exclusive ownership of the cloud-based object in response to the validation event, the cloud-based object having exclusivity of ownership such that a single license server can own the cloud-based object at any time (act 202). For example, license server 102 can send ownership request 111 to object management system 103 in response to validation event 104.

Ownership request 111 can be a request to assume ownership of object 106 (a cloud-based object).

5 **[0031]** Method 200 includes an act of the license server being granted exclusive ownership of the cloud-based object (act 203). For example, object management system 103 can grant ownership of object 106 to license server 102. Object management system 103 can formulate ownership grant 112 indicating ownership of object 106. Object management system 103 can send ownership grant 112 to license server 102.

[0032] Subsequently, license server 102 can access object 106 and corresponding properties, including name 107.

10 **[0033]** Method 200 includes an act of the license server using the unique name to establish the unique identifier as the identifier for the license server (act 204). For example, license server 102 can use name 107 to establish identifier 108 as the identifier for license server 102. Other components (e.g., applications) can then direct license requests corresponding to software license 121 to license server 102. When appropriate,
15 these other components can verify that license server 102 is in possession of identifier 108 (and is thus the license server for software license 121).

[0034] Embodiments of the invention include using a storage system that supports storage accounts referenced by globally unique account names. The storage system is accessible from a cloud environment where the license server is hosted. Inside each
20 storage account, there is an entity whose access right can only be owned by a single license server in the cloud at any time. By combining globally unique storage account names with an exclusive access right to the specific unique entity, a unique ID can be created that a single instance in the cloud can possess at any time.

[0035] Thus, a software license issued by an Independent Software Vendor (“ISV”) can use the unique storage account name to specify where it can be installed. The license
25 server in the cloud periodically checks the ownership of the access right to the specific unique entity in that storage account. Checking ownership helps insure that the license server is the only license server bound to the corresponding unique ID. Accordingly, the license cannot be repeatedly deployed on multiple license server instances.

30 **[0036]** Ownership of a unique ID can be secured through security mechanisms of the underlying storage system. That is, a license server cannot assume a unique storage account name as its unique ID without access rights to the specific unique entity within this account. Thus, it is possible to leverage reliability features of a cloud environment for

hosting a license server. For example, unlike hardware based IDs, storage account based IDs can be easily transferred from one cloud instance to another.

[0037] Some embodiments of the invention are implemented in a Windows Azure™ platform. These embodiments utilize a globally exclusive Azure storage account name with an exclusive mapping of a specific page blob within the storage account as a Windows Azure XDrive. A page blob is a collection of (e.g., 512-byte) pages used for random read and write operations. An xDrive enables a Windows Azure™ application to use existing Application Program Interfaces (“APIs”) to access a durable drive. This allows Windows Azure™ applications to mount a page blob as a drive letter (e.g., “X:”). The mapping is used to create an ID that one Azure instance can possess at any time.

[0038] An Azure storage account name can be a formal DNS name which is globally unique. Windows Azure XDrive permits the lease of a page blob formatted as a single New Technology File System (“NTFS”) volume Virtual Hard Drive (“VHD”) for mounting an XDrive to be exclusively owned by a single Azure instance. This makes an XDrive effectively a mutex for Azure instances. A software license can therefore use an Azure storage account name to identify its installation location. The license server periodically checks the validity of the XDrive mapping to make sure that it owns the ID, and terminates itself when it is no longer the owner.

[0039] Figure 3 illustrates another example computer architecture 300 that facilitates validating license servers in virtualized environments. As depicted, computer architecture 300 includes cloud environment 301 and storage system 311. Cloud environment 301 and storage system 311 can be connected to one another by a network connection.

[0040] Cloud environment 301 includes cloud license service instance 302. Cloud license service instance 302, storage account access information 303 and license server 304, which further includes enforcement plug-in 306. Storage system 311 includes storage account 312, which further includes unique entity 313.

[0041] As indicated by arrow 321, cloud license service instance 302 holds storage account access information 303 for storage account 312 (a storage account cloud license service instance 302 is bound to). As indicated by arrow 322, license server 304, through license server ID enforcement plug-in 306, locates unique entity 313 in storage account 312. License server 304 attempts to assume an exclusive access right to unique entity 313. If the attempt fails, cloud license service instance 302 can shut itself down. As indicted by arrow 323, license server 304 periodically checks ownership of the exclusive access right.

If ownership is ever lost, cloud license service instance 302 can attempt to recover it. If the attempt fails, cloud license service instance 302 can shut itself down.

[0042] Accordingly, embodiments of the invention leverage a set of features acquired or built in cloud computing environments to facilitate a software based solution providing uniqueness and immutability of a license server hosted in the cloud. Avoiding features of the underlying hardware systems results a much more flexible and reliable platform for hosting license servers. Features of a cloud storage service can be used to create a unique ID for a license server. Security and reliability of license servers hosted in a public cloud environment is also improved.

[0043] The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

CLAIMS

What is claimed:

1. At a cloud instance in a cloud computing environment, the cloud instance running on an underlying computer system that includes one or more processors and system memory, the cloud instance hosting a license server, a method for validating the license server as the target license server for a software license, the method comprising:

an act of detecting a validation event for the license server, the validation event indicating that the license server is to check status as the target license server for the software license, the software license installable at a license server having a unique identifier, the unique identifier based on a unique name associated with a cloud-based object within the cloud computing environment;

an act of license server requesting to assume exclusive ownership of the cloud-based object in response to the validation event, the cloud-based object having exclusivity of ownership such that a single license server can own the cloud-based object at any time;

an act of the license server being granted exclusive ownership of the cloud-based object; and

an act of the license server using the unique name to establish the unique identifier as the identifier for the license server.

2. The method as recited in claim 1, wherein the act of detecting a validation event comprises an act of detecting that the software license was stored within the cloud computing environment for installation at a license server having the unique identifier.

3. The method as recited in claim 1, wherein the act of detecting a validation event comprises an act of detecting that a specified period of time has elapsed since the status of the license server was last checked.

4. The method as recited in claim 1, wherein the act of license server requesting to assume exclusive ownership of the cloud-based object comprises an act of requesting to assume exclusive ownership of the cloud-based object associated with a storage location in the cloud computing environment.

5. The method as recited in claim 1, wherein the unique identifier is the unique name and wherein the unique name is the name of the storage location.

6. A computer program product for use at a cloud instance in a cloud computing environment, the cloud instance running on an underlying computer system that includes one or more processors and system memory, the cloud instance hosting a

license server, the computer program product for implementing a method for validating the license server as the target license server for a software license, the program product comprising one or more computer storage media having stored thereon computer-executable instructions that, when executed a processor, cause the cloud instance to perform the method, including the following:

detect a validation event for the license server, the validation event indicating that the license server is to check status as the target license server for the software license, the software license installable at a license server having a unique identifier, the unique identifier based on a unique name associated with a cloud-based object within the cloud computing environment;

request to assume exclusive ownership of the cloud-based object in response to the validation event, the cloud-based object having exclusivity of ownership such that a single license server can own the cloud-based object at any time;

be granted exclusive ownership of the cloud-based object; and

use the unique name to establish the unique identifier as the identifier for the license server.

7. The computer program product as recited in claim 6, wherein computer-executable instructions that, when executed, cause the cloud instance to detect a validation event comprises computer-executable instructions that, when executed, cause the cloud instance to detect that the software license was stored within the cloud computing environment for installation at a license server having the unique identifier.

8. The computer program product as recited in claim 6, wherein computer-executable instructions that, when executed, cause the cloud instance to detect a validation event comprise computer-executable instructions that, when executed, cause the cloud instance to detect that a specified period of time has elapsed since the status of the license server was last checked.

9. The computer program product as recited in claim 6, wherein computer-executable instructions that, when executed, cause the cloud instance to request to assume exclusive ownership of the cloud-based object comprise computer-executable instructions that, when executed, cause the cloud instance to request to assume exclusive ownership of the cloud-based object associated with a storage location in the cloud computing environment.

10. The computer program product as recited in claim 6, wherein the unique identifier is the unique name and wherein the unique name is the name of the storage location.

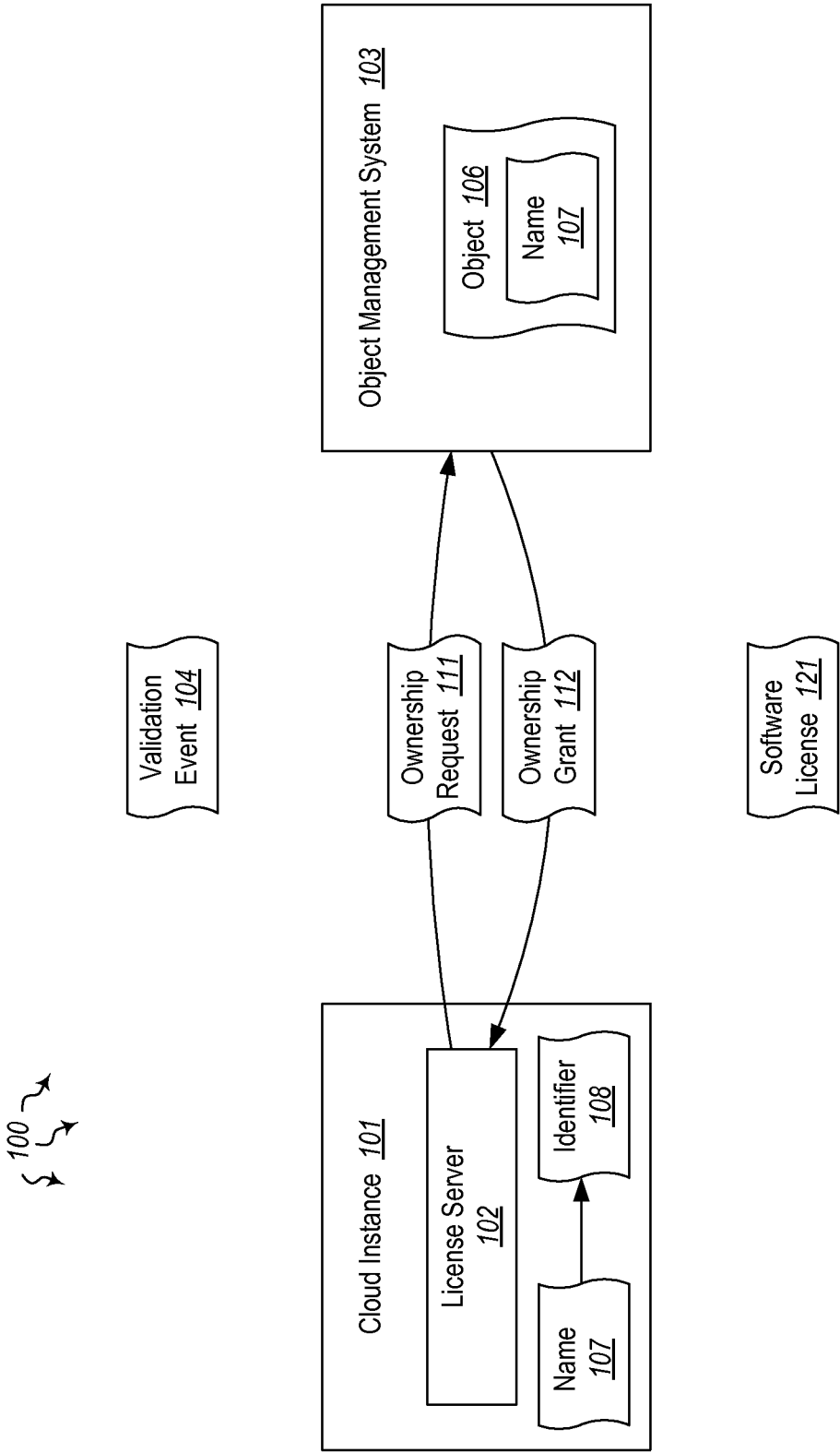
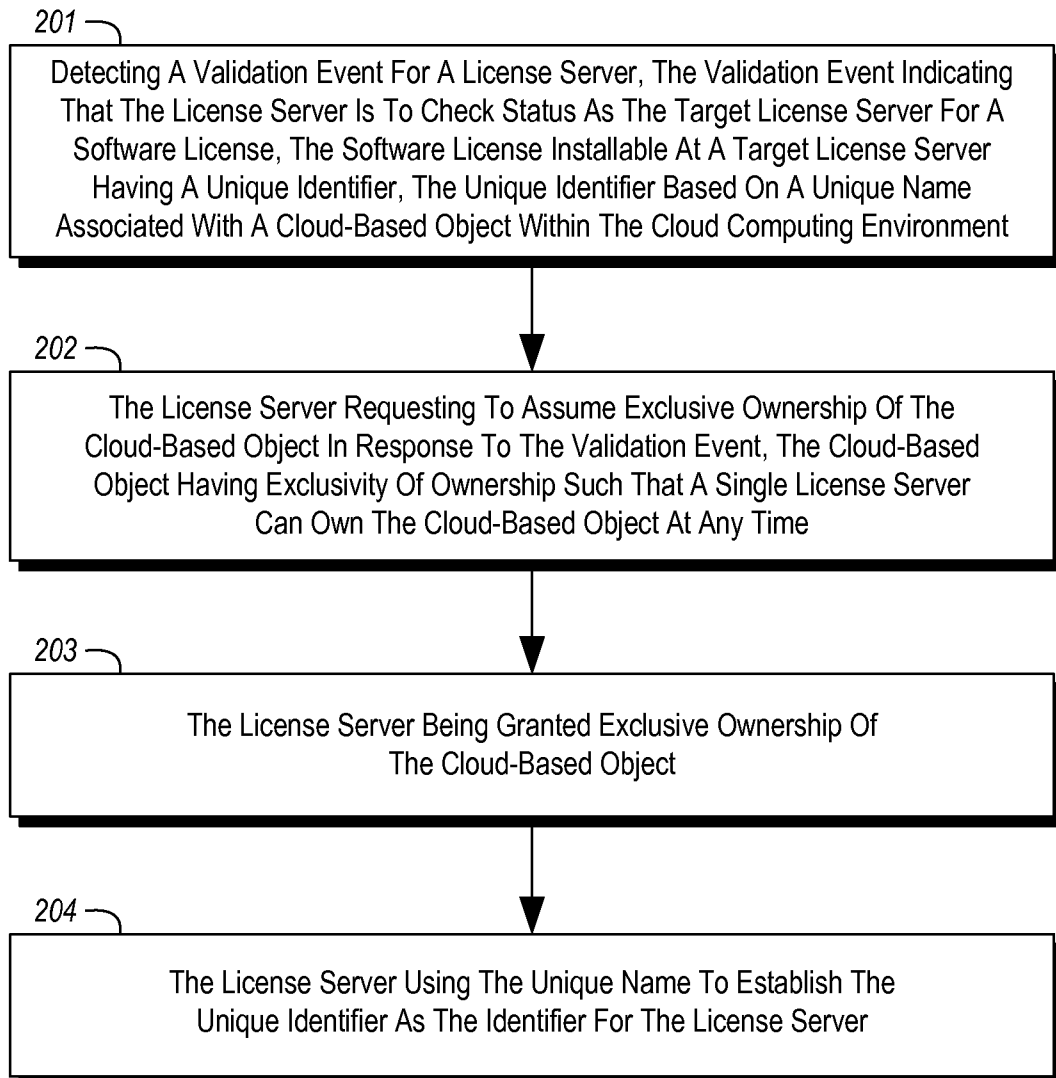


Figure 1

2 / 3

200**Figure 2**

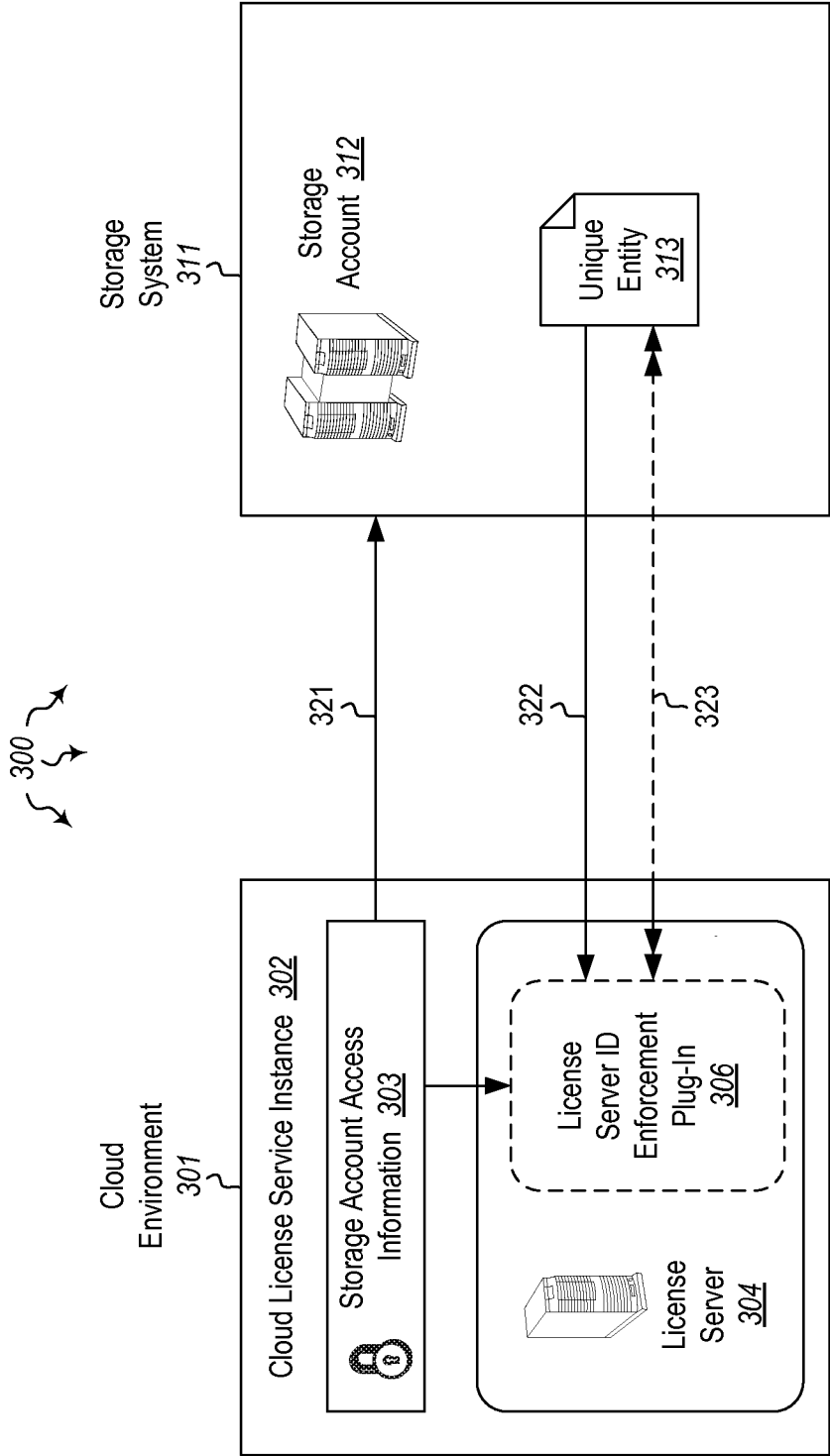


Figure 3

A. CLASSIFICATION OF SUBJECT MATTER**G06F 21/44(2013.01)i, G06F 15/16(2006.01)i**

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 21/44; G06F 21/24; G06F 15/173; G06F 21/20; G06F 17/30; G06F 12/14; H04L 9/32; G06F 9/455; G06F 15/16

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: license, server, cloud, object, unique, name, identifier, ownership, validation

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2012-0011244 A1 (ZHONGWEN ZHU) 12 January 2012 See paragraphs 11-12, 24, 31; and figure 3.	1-10
A	US 2009-0183010 A1 (PATRIK SCHNELL et al.) 16 July 2009 See paragraphs 22, 26, 29, 37, 40; and figure 1.	1-10
A	US 2009-0300607 A1 (JAMES MICHAEL FERRIS et al.) 03 December 2009 See paragraphs 11, 15, 23-27, 33-35; and figures 1-3.	1-10
A	KR 10-1107056 B1 (INITECH CO., LTD.) 25 January 2012 See paragraphs 6-8; and figure 1.	1-10
A	US 2011-0099616 A1 (LESZEK MAZUR et al.) 28 April 2011 See paragraphs 3, 12; and figure 1.	1-10
A	US 2010-0332456 A1 (ANAND PRAHLAD et al.) 30 December 2010 See paragraphs 63, 349; and figures 1, 16.	1-10



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of 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

09 July 2013 (09.07.2013)

Date of mailing of the international search report

10 July 2013 (10.07.2013)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City,
302-701, Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

BYUN Sung Cheal

Telephone No. +82-42-481-8262



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2013/027258

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0011244 A1	12/01/2012	WO 2012-004775 A2 WO 2012-004775 A3	12/01/2012 08/03/2012
US 2009-0183010 A1	16/07/2009	CN 101911087 A CN 101911087 B EP 2250608 A1 JP 2011-510539 A KR 10-2010-0110816 A RU 2010-129044 A WO 2009-091459 A1	08/12/2010 13/03/2013 17/11/2010 31/03/2011 13/10/2010 20/01/2012 23/07/2009
US 2009-0300607 A1	03/12/2009	US 8341625 B2	25/12/2012
KR 10-1107056 B1	25/01/2012	None	
US 2011-0099616 A1	28/04/2011	CN 102598577 A EP 2491673 A2 JP 2013-508854 A WO 2011-049784 A2 WO 2011-049784 A3	18/07/2012 29/08/2012 07/03/2013 28/04/2011 18/08/2011
US 2010-0332456 A1	30/12/2010	AU 2010-266433 B2 CA 2765624 A1 EP 2449477 A2 US 2010-0332401 A1 US 2010-0332454 A1 US 2010-0332479 A1 US 2010-0332818 A1 US 2010-0333116 A1 US 2013-024424 A1 US 8285681 B2 US 8407190 B2 WO 2011-002777 A2 WO 2011-002777 A3	21/02/2013 06/01/2011 09/05/2012 30/12/2010 30/12/2010 30/12/2010 30/12/2010 30/12/2010 24/01/2013 09/10/2012 26/03/2013 06/01/2011 21/04/2011