



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

(21) International Application Number:
PCT/US2009/034765

(22) International Filing Date:
20 February 2009 (20.02.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/030,909 22 February 2008 (22.02.2008) US

(71) Applicant and

(72) Inventor: **ETCHEGOYEN, Craig, S.** [US/US]; C/o
Uniloc USA Inc., 3333 Michelson Drive, Suite 600,
Irvine, CA 92612 (US).

(74) Agent: **PAIK, John, L.**; Connolly Bove Lodge & Hutz
LLP, P.O. Box 2207, Wilmington, DE 19899 (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, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG,
SK, SL, SM, ST, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,
ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ,
TM), European (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, SE, SI, SK, TR),
OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML,
MR, NE, SN, TD, TG).

Published:

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

(54) Title: LICENSE AUDITING FOR DISTRIBUTED APPLICATIONS

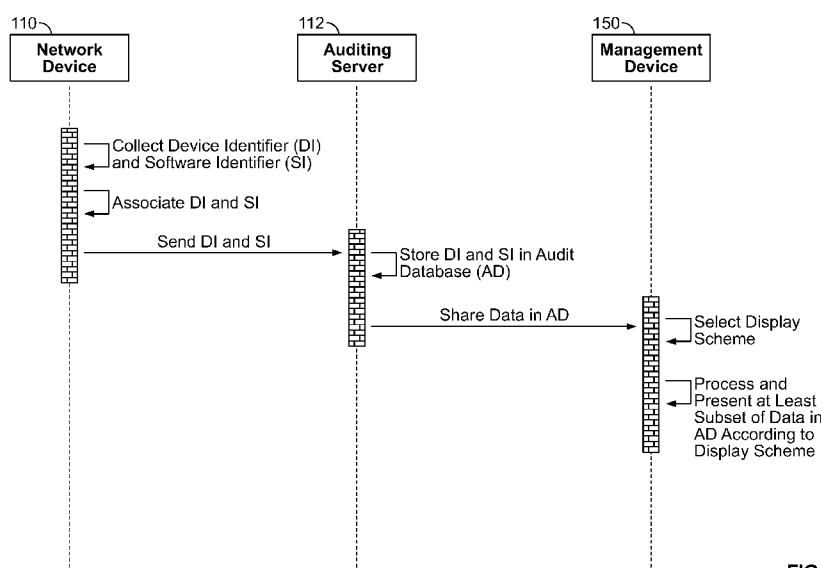


FIG. 13

(57) Abstract: The provided software application includes a module that determines a machine fingerprint of a client device at an appropriate time, such as during initial software load on the client. The fingerprint may comprise various machine-determinable measures of system configuration for the client. Each application copy may be associated with a serial number. A license host may collect serial number, fingerprint and/or IP address information from clients on which the application is installed. The host may generate a map of application installations, including geographic locations of installations and number of unique serial numbers per client in specified regions.

LICENSE AUDITING FOR DISTRIBUTED APPLICATIONS

Background of the Invention

Field of the Invention

[0001] The present invention is directed toward systems for auditing software licenses, and more particularly, to a system that interfaces with a user's device to measure the device hardware configuration and thereby generate a device identifier used to audit software on the device.

Description of the Related Art

[0002] Currently, there are limited ways to audit software licenses. At the same time, software piracy continues to grow at an alarming rate, particularly in emerging economies. In response, software companies have focused primarily on making it more difficult for would-be pirates to install a given software application illegally, such as a single copy of the application on multiple machines. Still, software pirates have found ways to bypass such security measures and install unauthorized software copies on multiple machines. Accordingly, it would be desirable to provide an auditing service for reliable software license authentication and to provide software owners with a measure of how many copies of their software have been legitimately registered and/or how many copies are pirated versions.

Summary of the Invention

[0003] The following presents a simplified summary of one or more embodiments in order to provide a basic understanding of such embodiments. This summary is not an extensive overview of all contemplated embodiments, and is intended to neither identify key or critical elements of all embodiments nor delineate the scope of any or all embodiments. Its sole purpose is to present some concepts of one or more embodiments in a simplified form as a prelude to the more detailed description that is presented later.

[0004] In accordance with one or more embodiments and corresponding disclosure thereof, various aspects are described in connection with auditing a license for a software running on a network device. In one embodiment, the device may include: a communication module for communicating with a server; at least one processor in operative communication

with the communication module; and a memory in operative communication with the at least one processor and comprising executable code for the at least one processor.

[0005] For example, a processor of the device may collect machine parameters of the device, the collected machine parameters comprising a combination of at least one user-configurable parameter and at least one non-user-configurable parameter. A processor (e.g., the same processor involved in collecting the machine parameters and/or a different processor) may generate a device identifier for the device based at least in part on the collected machine parameters. A processor may create an audit number based at least in part on the device identifier. A processor may instruct the communication module to transmit the audit number to an audit database for storage.

[0006] In related aspects, the memory of the device may include executable code for the at least one processor to: collect a software identifier of the software running on the device; and associate the software identifier with the device identifier to generate the audit number.

[0007] In further related aspects, the at least one processor may determine a geo-location code for the device (e.g., an Internet Protocol (IP) address); and associate the geo-location code with at least one of the software identifier and the device identifier to generate the audit number.

[0008] In accordance with other aspects of the embodiments described herein, there is provided a network device adapted to facilitate auditing of a license for a software running on the device. The device may include: a communication module for communicating with an auditing server and an authentication server; at least one processor in operative communication with the communication module; and a memory in operative communication with the at least one processor and comprising executable code for the at least one processor.

[0009] For example, a processor of the device may collect machine parameters of the device, the collected machine parameters comprising a combination of at least one user-configurable parameter and at least one non-user-configurable parameter. A processor may instruct the communication module to send the collected machine parameters to the auditing server to generate a device identifier for the device based at least in part on the collected machine parameters. A processor may, in response to receiving the device identifier from the server, create an audit number based at least in part on the device identifier. A processor may

instruct the communication module to transmit the audit number to at least one of the auditing server and the authentication server for storage in at least one audit database.

[0010] In related aspects, the memory may include executable code for the at least one processor to: collect a software identifier of the software running on the device; and associate the software identifier with the device identifier to generate the audit number. In the alternative, or in addition, the memory may include executable code for the at least one processor to: collect a software identifier of the software running on the network device; and instruct the communication module to send the software identifier to the auditing server to generate the device identifier based at least in part on the software identifier.

[0011] In further related aspects, the at least one processor may determine a geo-location code for the device; and associate the code with at least one of the device identifier and the software identifier to generate the audit number. In the alternative, or in addition, the at least one processor may send the code to the auditing server to generate the device identifier based at least in part on the code.

[0012] In accordance with other aspects of the embodiments described herein, there is provided a system for managing a license for a software running on one or more network devices. The system may include a communication module for accessing an audit database, the database comprising audit numbers for the one or more network devices, each audit number comprising a software identifier associated with a device identifier. The system may include: a display module; at least one processor in operative communication with the communication module, and the display module; and a memory in operative communication with the at least one processor and comprising executable code for the at least one processor.

[0013] For example, a processor of the system may (a) instruct the communication module to access the database and (b) process the audit numbers to, for example, sort the audit numbers according to at least one of activated license seats and unactivated license seats. A processor may select a display scheme for presenting the processed audit numbers, and instruct the display module to present the processed audit numbers according to the selected display scheme. In one approach, the processor may select the display scheme in response to a user choice entered on a user input module in operative communication with the at least one processor.

[0014] To the accomplishment of the foregoing and related ends, the one or more embodiments comprise the features hereinafter fully described and particularly pointed out in the claims. The following description and the annexed drawings set forth in detail certain illustrative aspects of the one or more embodiments. These aspects are indicative, however, of but a few of the various ways in which the principles of various embodiments may be employed and the described embodiments are intended to include all such aspects and their equivalents.

Brief Description of the Drawings

[0015] Figure 1A provides a block diagram of an exemplary system for auditing distributed software.

[0016] Figure 1B provides a block diagram of another exemplary system for auditing distributed software, wherein the system includes an authentication server.

[0017] Figure 2 illustrates the components of an exemplary device identifier.

[0018] Figures 3-12 illustrate exemplary display schemes for presenting data from the audit database regarding one or more network devices running a given software.

[0019] Figure 13 is a sequence diagram for the system of Figure 1A in accordance with an exemplary approach to auditing distributed software.

[0020] Figure 14 is a sequence diagram for the system of Figure 1B in accordance with another exemplary approach to auditing distributed software.

Detailed Description

[0021] The present invention addresses the need for an auditing service that provides reliable software license authentication and provides software owners with a measure of how many copies of their software have been legitimately registered and/or are pirated. Such an auditing service may be used alone, or in conjunction with other security/authentication measures.

[0022] The present technology provides for an improved system and method for auditing distributed software. In accordance with one aspect of the present technology, there is provided a system and method for authenticating software licenses. With reference to Figure 1A, there is provided an embodiment of a system having a plurality of network devices 110 that are in operative communication with an auditing server 112. While only one network device 110 is illustrated in Figures 1A-1B, it will be understood that a given system may comprise any number of network devices. The network device 110 may be, but is not limited to, a personal computer, a server computer, a laptop computer, a tablet computer, a personal digital assistant, a mobile phone, a wireless communication device, an onboard vehicle computer, or any other device capable of communication with a computer network.

[0023] The network device 110 may comprise a software 120 that requires a license to be authorized for use. The device 110 may also comprise an auditing tool or application 122. The auditing application 122 may be any program or application that collects identifying information regarding the network device 110 and/or any software (e.g., software 120) on the network device 110. The auditing application 122 may comprise a stand alone application or an applet running within a web browser on the device 110 (e.g., an applet comprising executable code for a Java Virtual Machine). The auditing application 122 may be embedded in or associated with another software application, including but not limited to software 120. For example, the auditing application 122 may be embedded in or associated with a tool bar of a software application, such as, for example, a web browser. The auditing application 122 may prompt the user to register with an online software registration service, or may run in the background with little or no interaction with the user of device 110.

[0024] The auditing application 122 may include a registration routine that collects information regarding network device 110 by checking a number of parameters which are expected to be unique to the network device environment. The parameters checked may include, for example, hard disk volume name, user name, device name, user password, hard disc initialization date, etc.. The collected information may include information that identifies the hardware comprising the platform on which the web browser runs, such as, for example, CPU number, or unique parameters associated with the firmware in use. The system information may further include system configuration information, such as amount of memory, type of processor, software or operating system serial number, etc.

[0025] Based on the collected information, the auditing application 122 may generate a device identifier 124 that is unique for the user computer 110. In the alternative, or in addition, the application 122 may gather and send the device parameters to a remote server, such as auditing server 112, which in turn generates the device identifier 124. The device identifier 124 may be stored in a hidden directory of the device 110 and/or at a remote location, such as the auditing server 112. The device identifier 124 may incorporate the device's IP address and/or other geo-location code to add another layer of specificity to device's unique identifier.

[0026] It is noted that an application (e.g., auditing application 122) running on the network device or otherwise having access to the network device's hardware and file system may generate a unique device identifier (e.g., device identifier 124) using a process that operates on data indicative of the network device's configuration and hardware. The device identifier may be generated using a combination of user-configurable and non-user-configurable machine parameters as input to a process that results in the device identifier, which may be expressed in digital data as a binary number. Each machine parameter is data determined by a hardware component, software component, or data component specific to the device that the unique identifier pertains to. Machine parameters may be selected based on the target device system configuration such that the resulting device identifier has a very high probability (e.g., greater than 99.999%) of being unique to the target device. In addition, the machine parameters may be selected such that the device identifier includes at least a stable unique portion up to and including the entire identifier, that has a very high probability of remaining unchanged during normal operation of the target device. Thus, the resulting device identifier should be highly specific, unique, reproducible and stable as a result of properly selecting the machine parameters.

[0027] The application for generating the device identifier may also operate on the collected parameters with one or more algorithms to generate the device identifier. This process may include at least one irreversible transformation, such as, for example, a cryptographic hash function, such that the input machine parameters cannot be derived from the resulting device identifier. Each device identifier, to a very high degree of certainty, cannot be generated except by the suitably configured application operating or otherwise having had access to the same field security device for which the device identifier was first

generated. Conversely, each identifier, again to a very high degree of certainty, can be successfully reproduced by the suitably configured application operating or otherwise having access to the same field security device on which the identifier was first generated.

[0028] The application may operate by performing a system scan to determine a present configuration of the field security device. The application may then select the machine parameters to be used as input for generating the unique device identifier. Selection of parameters may vary depending on the system configuration. Once the parameters are selected, the application may generate the identifier.

[0029] Further, generating the device identifier may also be described as generating a device fingerprint and may entail the sampling of physical, non-user configurable properties as well as a variety of additional parameters such as uniquely generated hashes and time sensitive values. Physical device parameters available for sampling may include, for example, unique manufacturer characteristics, carbon and silicone degradation and small device failures.

[0030] The process of measuring carbon and silicone degradation may be accomplished by measuring a chip's ability to process complex mathematical computations, and its ability to respond to intensive time variable computations. These processes measure how fast electricity travels through the carbon. Using variable offsets to compensate for factors such as heat and additional stresses placed on a chip during the sampling process allows for each and every benchmark to reproduce the expected values. During a standard operating lifetime, the process of passing electricity through the various switches causes a computer chip to degrade. These degradations manifest as gradually slower speeds that extend the processing time required to compute various benchmarking algorithms.

[0031] In addition to the chip benchmarking and degradation measurements, the process for generating a device identifier may include measuring physical, non-user-configurable characteristics of disk drives and solid state memory devices. Each data storage device has a large variety of damage and unusable data sectors that are nearly unique to each physical unit. The ability to measure and compare values for damaged sectors and data storage failures provides a method for identifying storage devices.

[0032] Device parameter sampling, damage measurement and chip benchmarking make up just a part of device fingerprinting technologies described herein. These tools may be further

extended by the use of complex encryption algorithms to convolute the device identifier values during transmission and comparisons. Such encryption processes may be used in conjunction with random sampling and key generations.

[0033] The device identifier may be generated by utilizing machine parameters associated with one or more of the following: machine model; machine serial number; machine copyright; machine ROM version; machine bus speed; machine details; machine manufacturer; machine ROM release date; machine ROM size; machine UUID; and machine service tag.

[0034] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: CPU ID; CPU model; CPU details; CPU actual speed; CPU family; CPU manufacturer; CPU voltage; and CPU external clock.

[0035] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: memory model; memory slots; memory total; and memory details.

[0036] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: video model; video details; display model; display details; audio model; and audio details.

[0037] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: network model; network address; Bluetooth address; Blackbox model (including IDE and SCSI); Blackbox serial; Blackbox details; Blackbox damage map; Blackbox volume name; NetStore details; and NetStore volume name.

[0038] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: optical model; optical serial; optical details; keyboard model; keyboard details; mouse model; mouse details; printer details; and scanner details.

[0039] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: baseboard manufacturer; baseboard product name; baseboard version; baseboard serial number; and baseboard asset tag.

[0040] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: chassis manufacturer; chassis type; chassis version; and chassis serial number.

[0041] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: IDE controller; SATA controller; RAID controller; and SCSI controller.

[0042] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: port connector designator; port connector type; port connector port type; and system slot type.

[0043] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: cache level; cache size; cache max size; cache SRAM type; and cache error correction type.

[0044] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: fan; PCMCIA; modem; portable battery; tape drive; USB controller; and USB hub.

[0045] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: device model; device model IMEI; device model IMSI; and device model LCD.

[0046] The device identifier may also be generated by utilizing machine parameters associated with one or more of the following: wireless 802.11; webcam; game controller; silicone serial; and PCI controller.

[0047] With reference to Figure 2, in one embodiment, the device identifier 124 may include two components – namely, a variable key portion 126 and a system key portion 128. The variable key portion 126 may be generated at the time of registration of network device 110 by reference to a variable platform parameter, such as via reference to system time information, although other parameters which are variable may be utilized in other embodiments. The system key portion 128 may include the above described parameters expected to be unique to the device 110, such as, for example, hard disk volume name, user

name, computer name, user password, hard disc initialization date, or combinations thereof. Portions 126 and/or 128 may be combined with the IP address and/or other platform parameters of the device 110. Further details regarding device identifiers 124, such as machine fingerprints and parameters expected to be unique for the a given computer, can be found in U.S. Application No. 08/124,718, filed September 21, 1993, titled "System for Software Registration," now issued as U.S. Patent No. 5,490,216, which application is specifically incorporated herein, in its entirety, by reference. It is noted that device identifiers, or portions thereof, may be encrypted to add an additional layer of specificity and security.

[0048] With reference once again to Figure 1A, the auditing application 122 may also include a registration routine that collects or receives information regarding the software 120 on device 110 by checking information which is expected to be unique to software 120, such as, for example, the software serial number. The collected software identifier may include the software serial number, product identification number, product key, etc. The collected software identifier may include information regarding where the software was sold or distributed, who the buyers, sellers, and/or distributors were, which stores the software was sold in, etc. It is noted that the software identifier may be unique to particular copy of software, such as when the software is licensed to a single user. In the alternative, or in addition, the software identifier may be unique to particular type or group of a software, such as when the software is licensed to a defined group of users.

[0049] The embodiments described herein comprise an auditing application 122 that collects the software identifier 130 for software 120; however, it will be understood that the systems and components described herein can be adapted to collect one or more types of software identifiers for a plurality of software applications. The software identifier 130 may be stored in a hidden directory of the device 110 and/or at a remote location, such as the auditing server 112.

[0050] The auditing application 122 may also include a registration routine that collects or receives information regarding the geo-location code 140 of the device 110. The geo-locator 140 may comprise the IP address or the like of the device 110.

[0051] Auditing application 122 may electronically send the device identifier 124 and the software identifier 130 to the auditing server 112. In the alternative, or in addition, a geo-

location code 140, such as the IP address of the device 110, may be associated with the device identifier 124 and/or the software identifier 130 and may be sent to the auditing server 112, such as via a secured network connection. The auditing server 112 may encrypt and store the data, such as the device identifier 124, the software identifier 130, and/or the geo-location code 140, received from the network device 110. The auditing server 112 may receive such data from a plurality of network devices and store the received data in an audit database 114.

[0052] In one embodiment, the auditing application 122 may generate an audit number 142 by associating the software identifier 130 with the device identifier 124 and/or geo-location code 140, and may send the generated audit number 142 to the auditing server 112. In another embodiment, the application 122 may send the device identifier 124, the software identifier 130, and/or the geo-location code 140 to the server 112 in a piecemeal manner. The server 112 may in turn generate the audit number 142. The auditing server 112 may receive or generate audit numbers from a plurality of network devices and store the received audit numbers in the audit database 114.

[0053] It is noted that the audit number 142 may be generated from the device identifier 124, the software identifier 130, and/or the geo-location code 140 via any number of suitable approaches. For example, the software identifier 130 may be concatenated or linked with the device identifier 124 and/or geo-location code 140. It is also noted that the audit number 142 may be stored in a hidden directory of the device 110 and/or at a remote location, such as the auditing server 112. It is further noted that the device identifier 124, the software identifier 130, and/or the geo-location code 140 may at a later time be extracted from the audit number 142.

[0054] When a user of a network device, including but not limited to network device 110, installed with auditing application 122, attempts to run the software 120, the auditing application 122 in response may transmit the software identifier 130 associated with the device identifier 124 and/or the geo-location code 140 (or an audit number 142 generated from such data) to the auditing server 112, which in turn may store the received data in the audit database 114.

[0055] With reference to Figure 1B, there is provided an embodiment of a system that further comprises an optional authentication server 116 that is in operative communication

with the auditing server 112. When a given user tries to run software 120 on his/her network device, the authentication server 116 may access the audit database 114 on auditing server 112 to determine whether to allow his/her of the software 120. The authentication server 116 may receive/access the license terms for a particular software from the auditing server 112 or another server or network device.

[0056] In one embodiment, the authentication server 116 may disallow use of the software 120 beyond a defined maximum number of allowed users or seats (which may be defined by the software license). The server 116 may analyze the data in the audit database 114 and determine how many seats are currently utilizing software 120. If the number of currently allowed seats meets or exceeds the maximum number of allowed seats, the server 116 may throttle or disallow the use of software 120 by more seats; otherwise, the server may allow the use of the software 120. In another embodiment, the authentication server 116 may throttle or disallow use of the software 120 if its software identifier is already associated with a different device identifier and/or a different IP address in the audit database 114; otherwise, the server 116 may allow use of the software 120. The authentication server 116 may collect data regarding the instances of allowed and disallowed software use, and may share such data with the auditing server 112.

[0057] With reference to the embodiments of Figures 1A and 1B, the auditing server 112 may be in operative communication with a management device 150, which may be any device capable of communication with a computer network, such as, for example, a personal computer, a server computer, a laptop computer, a tablet computer, a personal digital assistant, a mobile phone, or a wireless communication device. The management device 150 may comprise a management application 152, which may be any program or application, such as a stand alone application or an application that is embedded or associated with another software application, such as an applet running within a web browser on the device 150.

[0058] The management application 152 may be adapted to allow a user, such as, for example, a software manufacturer or distributor, to view the data collected and stored in the audit database 114 of the audit server 112. The present embodiment will be described in the context of a software manufacturer utilizing the management application 152. However, it will be understood that any user of the management device may utilize the management application 152.

[0059] The management application 152 may present the data in the audit database 114 in a manner that allows its user to better understand how its software is being used, legitimately or otherwise. The information organized and presented according to one or more display schemes of the application 152 may allow a software manufacturer to better understand software consumer behaviors and habits, which in turn may allow the manufacturers to adjust or modify their licensing rules to comport with their business goals.

[0060] The management application 152 may be adapted to process and/or present at least a subset of the data in the audit database 114 according to one or more display schemes. The display schemes may be predefined or presented for selection by the software manufacturer. The data in the audit database 114 (i.e., the audit numbers 142 and components thereof for network devices 110) may be organized or sorted by the number of activated/unactivated seats, license seat trends, activations trends, piracy curves, etc., as shown in Figures 3-12. It is noted that the data in the audit database 114 may be organized, processed, and processed by the management device 152 and/or the auditing server 112.

[0061] The exemplary display scheme of Figure 3 provides an activation seats trend, which shows the number of activated seats over a period of time for three types of software 310, 320, and 330 (e.g., corresponding to “2D Sketch and Plan,” “3D Floorplan” and “Home Plan Pro,” respectively). The data presented according to the display scheme of Figure 3 may allow one to determine licensing trends and correlate marketing and sales efforts to activation events.

[0062] The exemplary display scheme of Figure 4 provides the license seat trend 410, the activation seats trend 420, the activated-per-licensed-seat trend 430, and the activations trend 440 for three types of software (e.g., 310, 320, 330 in Figure 3).

[0063] The exemplary display scheme of Figure 5 provides a piracy heat map which may allow one to visualize the extent/level of piracy in different geographic regions. It is noted that such piracy heat maps and other geographically based summaries may be based at least in part on the geo-location codes associated with the corresponding software identifiers. The data presented on the piracy heat map may be filtered by piracy, activations, number of seats or licenses sold, etc. The display scheme may further comprise a pivot table or the like to supplement the piracy heat map.

[0064] The exemplary display scheme of Figure 6 presents the license seat utilization (activated vs. deactivated seats) summary 610 and the activation trend (activations vs. deactivations) summary 620. The exemplary display scheme of Figure 7 presents the license seat trend 710, the activation seats trend 720, the activated-per-licensed-seat trend 730, and the activation trend 740 for three types of software (e.g., “Escape from Alcatraz,” “Marks vs. Ninjas II” and “Revenge of the Pirates”).

[0065] The exemplary display scheme of Figure 8 presents activated-versus-unactivated pie charts 810, the licensee seat trend 820, and the activation seats trend 830. Similarly, the exemplary display scheme of Figure 9 presents activated-versus-unactivated pie charts 910, the licensee seat trend 920, and the activation seats trend 930 for five types of software (e.g., “2D Sketch and Plan,” “3D Floorplan,” “Design CAD,” “Home Plan Pro” and “IC Digital Design”). The presentation of data according to the display schemes of Figures 8 and 9 may make it possible to identify accounts that are at or above their licensed seats. In addition, the data presented in Figures 8 and 9 make it possible to identify licenses that are not being utilized and to properly allocate resources to ensure continued revenue streams.

[0066] The exemplary display scheme of Figure 10 presents piracy curves 1010, the top ten abusing countries summary 1020, and the pirated activations summary 1030. It is again noted that such geographically based summaries may be based at least in part on the geo-location codes associated with the corresponding software identifiers.

[0067] It is noted that the data in the audit database 114 may be used to model or simulate the effect of certain actions taken by the software manufacturer. The information generated by such licensing models or simulators may also be based in part on other trends/data. For example, the models may be based in part on historical, current, and/or forecast trends/data associated with a particular software identifier. Similarly, the models may be based in part on trends/data associated with a particular class or types of software and/or trends/data for the software industry. The models may be based in part on trends/data associated with other industries, such as, for example, the hardware industry. For example, the management application 152 may simulate the effect of throttling or disallowing the further use of a given software. The exemplary display scheme of Figure 11 summarizes the effect of such throttling by displaying the throttling-effectiveness summary 1110 and the throttling-rule-impact summary 1120. Similarly, the exemplary display scheme of Figure 12 presents the throttling-

effectiveness summary 1210 and the throttling-rule-impact summary 1220. The data presented according to the display schemes of Figures 11 and 12 may include data generated by what-if simulators or models, which in turn may allow one to better understand how the software licenses are consumed.

[0068] It will be understood that the described system for auditing software usage by network device users can comprise any number of components or modules adapted to perform various functions or tasks. For example, with reference to Figure 13, there is provided one embodiment of a software license auditing system wherein an auditing application on a network device 110 may collect information regarding the device 110 by checking hardware parameters expected to be unique to the network device environment. The auditing application may generate a device identifier based on the collected hardware information. In the alternative, or in addition, the auditing application may provide the collected hardware information to another device or server, which in turn may generate the hardware identifier.

[0069] The auditing application may collect or receive information regarding a given software on the network device 110, such as, for example, a software serial number. The auditing application may optionally collect or receive information regarding collects a geo-location code 140 of the device 110, such as, for example, the IP address for the device 110.

[0070] The auditing application may associate the software identifier with the hardware identifier and/or the geo-location code, and may provide such data to an auditing server 112. The auditing server 112 may receive such data from a plurality of network devices and store the received data in an audit database.

[0071] The auditing server 112 share the data in the audit database with a management device 150. The management device 150 may comprise a management application that selects a display scheme for presenting data in the audit database. The management application may automatically determine the appropriate display scheme, or may allow the management device user to select a display scheme from a menu or list. The management application may process and/or present at least a subset of the data in the audit database according to the selected display scheme.

[0072] With reference to Figure 14, there is provided another embodiment of an software license auditing system that is similar to the embodiment of Figure 13, but further comprises

an optional authentication server 116. The authentication server 116 may detect when a network device user is trying to run a given software on his/her network device. The authentication server 116 may receive/access and use the data in the audit database of the auditing server 112 to determine whether to allow the attempted use the given software. The authentication server 116 may further receive/access the license terms for a particular software from the auditing server 112 or another server or network device. In one approach, the server 116 may send a throttle or disallow use command to the network device if the number of currently allowed seats meets or exceeds the maximum number of allowed seats for the given software, which may be defined by the software license terms. In another approach, the server 116 may send a throttle or disallow command to the network device if software identifier for the given software is already associated with a different device identifier and/or a different IP address in the audit database. Otherwise, the server 116 may send an allowed use command to the network device to allow use of the given software.

[0073] In accordance with one or more aspects of the embodiments described herein, there are provided techniques for auditing licenses for software running on one or more network devices. In one embodiment, there is provided a network device comprising: a communication module for communicating with a server (e.g., audit database); at least one processor in operative communication with the communication module; and a memory in operative communication with the at least one processor and comprising executable code for the at least one processor to perform a number of steps. For example, the at least one processor may: collect machine parameters of the device, the collected machine parameters comprising a combination of at least one user-configurable parameter and at least one non-user-configurable parameter; generate a device identifier for the device based at least in part on the collected machine parameters; create an audit number based at least in part on the device identifier; and instruct the communication module to transmit the audit number to an audit database for storage.

[0074] In related aspects, the memory may comprise executable code for the at least one processor to: collect a software identifier of the software running on the device; and associate the software identifier with the device identifier to generate the audit number. The software identifier may comprise: a software serial number; information regarding a geographic region

associated with the software; and/or information regarding at least one of a seller, a buyer, and a distributor of the software.

[0075] In further related aspects, the at least one processor may associate the software identifier with the device identifier by concatenating the software identifier with the device identifier. In the alternative, or in addition, the at least one processor may associate the software identifier with the device identifier by linking the software identifier with the device identifier.

[0076] In yet further related aspects, the at least one processor may: determine a geo-location code for the device; and associate the geo-location code with at least one of the software identifier and the device identifier to generate the audit number. The geo-location code comprises an Internet Protocol (IP) address of the device.

[0077] In other aspects, the at least one processor may generate the device identifier by implementing at least one irreversible transformation (e.g., a cryptographic hash function) such that the collected machine parameters cannot be derived from the device identifier.

[0078] In accordance with one or more aspects of the embodiments described herein, there is provided another embodiment of a network device adapted to facilitate auditing of a license for a software running on the device. The device may comprise: a communication module for communicating with an auditing server and an authentication server; at least one processor in operative communication with the communication module; and a memory in operative communication with the at least one processor and comprising executable code for the at least one processor to perform a number of tasks. For example, the at least one processor may: collect machine parameters of the device, the collected machine parameters comprising a combination of at least one user-configurable parameter and at least one non-user-configurable parameter; instruct the communication module to send the collected machine parameters to the auditing server to generate a device identifier for the device based at least in part on the collected machine parameters; in response to receiving the device identifier from the server, create an audit number based at least in part on the device identifier; and instruct the communication module to transmit the audit number to at least one of the auditing server and the authentication server for storage in at least one audit database.

[0079] In related aspects, the memory may further comprise executable code for the at least one processor to: collect a software identifier of the software running on the device; and associate the software identifier with the device identifier to generate the audit number.

[0080] In further related aspects, the memory may further comprise executable code for the at least one processor to: collect a software identifier of the software running on the network device; and instruct the communication module to send the software identifier to the auditing server to generate the device identifier based at least in part on the software identifier.

[0081] In other related aspects, the at least one processor may: determine a geo-location code (e.g., IP address) or the device; and associate the code with at least one of the device identifier and the software identifier to generate the audit number. In the alternative, or in addition, the at least one processor may: determine a geo-location code for the device; and instruct the communication module to send the code to the auditing server to generate the device identifier based at least in part on the code.

[0082] In accordance with one or more aspects of the embodiments described herein, there is provided a system for managing a license for a software running on one or more network devices, comprising: a communication module for accessing an audit database, the database comprising audit numbers for the one or more network devices, each audit number comprising a software identifier associated with a device identifier, each device identifier being generated from a combination of user-configurable and non-user-configurable machine parameters for a given network device; a display module; at least one processor in operative communication with the communication module, and the display module; and a memory in operative communication with the at least one processor and comprising executable code for the at least one processor. For example, the at least one processor may: instruct the communication module to access the database; process the audit numbers to sort the audit numbers according to at least one of activated license seats and unactivated license seats; select a display scheme for presenting the processed audit numbers; and instruct the display module to present the processed audit numbers according to the selected display scheme. The at least one processor may select the display scheme in response to a user choice entered on a user input module in operative communication with the at least one processor.

[0083] In related aspects, the at least one processor may sort the audit numbers according to at least one of authorized software copies and pirated software copies. In the alternative, or in addition, the at least one processor may sort the audit numbers according to geographic data regarding the software.

[0084] In further related aspects, the at least one processor may identify a trend with respect to at least one of the activated license seats and the unactivated license seats. The at least one processor may identify a trend with respect to at least one of authorized software copies and pirated software copies. The at least one processor identifies a trend with respect to geographic data regarding the software. The at least one processor may instruct the display module to display the processed audit numbers as a piracy heat map or the like.

[0085] While the present invention has been illustrated and described with particularity in terms of preferred embodiments, it should be understood that no limitation of the scope of the invention is intended thereby. Features of any of the foregoing methods and devices may be substituted or added into the others, as will be apparent to those of skill in the art. It should also be understood that variations of the particular embodiments described herein incorporating the principles of the present invention will occur to those of ordinary skill in the art and yet be within the scope of the invention.

[0086] As used in this application, the terms “component,” “module,” “system,” and the like are intended to refer to a computer-related entity, either hardware, firmware, a combination of hardware and software, software, or software in execution. For example, a component can be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a computing device and the computing device can be a component. One or more components can reside within a process and/or thread of execution and a component can be localized on one computer and/or distributed between two or more computers. In addition, these components can execute from various computer readable media having various data structures stored thereon. The components can communicate by way of local and/or remote processes such as in accordance with a signal having one or more data packets (e.g., data from one component interacting with another component in a local system, distributed system, and/or across a network such as the Internet with other systems by way of the signal).

[0087] It is understood that the specific order or hierarchy of steps in the processes disclosed herein in an example of exemplary approaches. Based upon design preferences, it is understood that the specific order or hierarchy of steps in the processes may be rearranged while remaining within the scope of the present disclosure. The accompanying method claims present elements of the various steps in sample order, and are not meant to be limited to the specific order or hierarchy presented.

[0088] Moreover, various aspects or features described herein can be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques. The term “article of manufacture” as used herein is intended to encompass a computer program accessible from any computer-readable device, carrier, or media. For example, computer-readable media can include but are not limited to magnetic storage devices (e.g., hard disk, floppy disk, magnetic strips, etc.), optical disks (e.g., compact disk (CD), digital versatile disk (DVD), etc.), smart cards, and flash memory devices (e.g., Erasable Programmable Read Only Memory (EPROM), card, stick, key drive, etc.). Additionally, various storage media described herein can represent one or more devices and/or other machine-readable media for storing information. The term “machine-readable medium” can include, without being limited to, wireless channels and various other media capable of storing, containing, and/or carrying instruction(s) and/or data.

[0089] Those skilled in the art will further appreciate that the various illustrative logical blocks, modules, circuits, methods and algorithms described in connection with the examples disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, methods and algorithms have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present invention.

WHAT IS CLAIMED IS:

1. A network device adapted to facilitate auditing of a license for a software running on the device, comprising:
 - a communication module for communicating with a server;
 - at least one processor in operative communication with the communication module; and
 - a memory in operative communication with the at least one processor and comprising executable code for the at least one processor to:
 - collect machine parameters of the device, the collected machine parameters comprising a combination of at least one user-configurable parameter and at least one non-user-configurable parameter;
 - generate a device identifier for the device based at least in part on the collected machine parameters;
 - create an audit number based at least in part on the device identifier;
 - and
 - instruct the communication module to transmit the audit number to an audit database for storage.
2. The device of Claim 1, wherein the memory further comprises executable code for the at least one processor to:
 - collect a software identifier of the software running on the device; and
 - associate the software identifier with the device identifier to generate the audit number.
3. The device of Claim 2, wherein the software identifier comprises a software serial number.
4. The device of Claim 2, wherein the software identifier comprises information regarding a geographic region associated with the software.
5. The device of Claim 2, wherein the software identifier comprises information regarding at least one of: a seller; a buyer; and a distributor of the software.

6. The device of Claim 2, wherein the at least one processor associates the software identifier with the device identifier by concatenating the software identifier with the device identifier.

7. The device of Claim 2, wherein the at least one processor associates the software identifier with the device identifier by linking the software identifier with the device identifier.

8. The device of Claim 2, wherein the at least one processor:
determines a geo-location code for the device; and
associates the geo-location code with at least one of the software identifier and the device identifier to generate the audit number.

9. The device of Claim 8, wherein the geo-location code comprises an Internet Protocol (IP) address of the device.

10. The device of Claim 1, wherein the at least one processor generates the device identifier by implementing at least one irreversible transformation such that the collected machine parameters cannot be derived from the device identifier.

11. The device of Claim 10, wherein the at least one irreversible transformation comprises a cryptographic hash function.

12. The device of Claim 1, wherein the server comprises the audit database.

13. The device of Claim 1, wherein the machine parameters comprise information regarding at least one of: machine model; machine serial number; machine copyright; machine ROM version; machine bus speed; machine details; machine manufacturer; machine ROM release date; machine ROM size; machine UUID; and machine service tag.

14. The device of Claim 1, wherein the machine parameters comprise information regarding at least one of: CPU ID; CPU model; CPU details; CPU actual speed; CPU family; CPU manufacturer; CPU voltage; and CPU external clock.

15. The device of Claim 1, wherein the machine parameters comprise information regarding at least one of: network model; network address; Bluetooth address; Blackbox

model (including IDE and SCSI); Blackbox serial; Blackbox details; Blackbox damage map; Blackbox volume name; NetStore details; and NetStore volume name.

16. The device of Claim 1, wherein the machine parameters comprise information regarding at least one of: optical model; optical serial; optical details; keyboard model; keyboard details; mouse model; mouse details; printer details; and scanner details.

17. The device of Claim 1, wherein the machine parameters comprise information regarding at least one of: baseboard manufacturer; baseboard product name; baseboard version; baseboard serial number; and baseboard asset tag.

18. The device of Claim 1, wherein the machine parameters comprise information regarding at least one of: chassis manufacturer; chassis type; chassis version; and chassis serial number.

19. The device of Claim 1, wherein the machine parameters comprise information regarding at least one of: IDE controller; SATA controller; RAID controller; and SCSI controller.

20. The device of Claim 1, wherein the machine parameters comprise information regarding at least one of: port connector designator; port connector type; port connector port type; and system slot type.

21. The device of Claim 1, wherein the machine parameters comprise information regarding at least one of: cache level; cache size; cache max size; cache SRAM type; and cache error correction type.

22. The device of Claim 1, wherein the machine parameters comprise information regarding at least one of: fan; PCMCIA; modem; portable battery; tape drive; USB controller; and USB hub.

23. The device of Claim 1, wherein the machine parameters comprise information regarding at least one of: device model; device model IMEI; device model IMSI; and device model LCD.

24. The device of Claim 1, wherein the machine parameters comprise information regarding at least one of: wireless 802.11; webcam; game controller; silicone serial; and PCI controller.

25. A network device adapted to facilitate auditing of a license for a software running on the device, comprising:

- a communication module for communicating with an auditing server and an authentication server;

- at least one processor in operative communication with the communication module; and

- a memory in operative communication with the at least one processor and comprising executable code for the at least one processor to:

- collect machine parameters of the device, the collected machine parameters comprising a combination of at least one user-configurable parameter and at least one non-user-configurable parameter;

- instruct the communication module to send the collected machine parameters to the auditing server to generate a device identifier for the device based at least in part on the collected machine parameters;

- in response to receiving the device identifier from the server, create an audit number based at least in part on the device identifier; and

- instruct the communication module to transmit the audit number to at least one of the auditing server and the authentication server for storage in at least one audit database.

26. The device of Claim 25, wherein the memory further comprises executable code for the at least one processor to:

- collect a software identifier of the software running on the device; and
- associate the software identifier with the device identifier to generate the audit number.

27. The device of Claim 25, wherein the memory further comprises executable code for the at least one processor to:

- collect a software identifier of the software running on the network device; and
- instruct the communication module to send the software identifier to the auditing server to generate the device identifier based at least in part on the software identifier.

28. The device of Claim 27, wherein the at least one processor:

- determines a geo-location code for the device; and
- associates the code with at least one of the device identifier and the software identifier to generate the audit number.

29. The device of Claim 28, wherein the code comprises an Internet Protocol (IP) address of the device.

30. The device of Claim 25, wherein the at least one processor:

- determines a geo-location code for the device; and
- instructs the communication module to send the code to the auditing server to generate the device identifier based at least in part on the code.

31. An system for managing a license for a software running on one or more network devices, comprising:

- a communication module for accessing an audit database, the database comprising audit numbers for the one or more network devices, each audit number comprising a software identifier associated with a device identifier, each device identifier being generated from a combination of user-configurable and non-user-configurable machine parameters for a given network device;

- a display module;

- at least one processor in operative communication with the communication module, and the display module; and

- a memory in operative communication with the at least one processor and comprising executable code for the at least one processor to:

- instruct the communication module to access the database;

- process the audit numbers to sort the audit numbers according to at least one of activated license seats and unactivated license seats;

- select a display scheme for presenting the processed audit numbers; and

- instruct the display module to present the processed audit numbers according to the selected display scheme.

32. The system of Claim 31, wherein the at least one processor selects the display scheme in response to a user choice entered on a user input module in operative communication with the at least one processor.

33. The system of Claim 31, wherein the at least one processor sorts the audit numbers according to at least one of authorized software copies and pirated software copies.

34. The system of Claim 31, wherein the at least one processor sorts the audit numbers according to geographic data regarding the software.

35. The system of Claim 31, wherein the at least one processor identifies a trend with respect to at least one of the activated license seats and the unactivated license seats.

36. The system of Claim 31, wherein the at least one processor identifies a trend with respect to at least one of authorized software copies and pirated software copies.

37. The system of Claim 31, wherein the at least one processor identifies a trend with respect to geographic data regarding the software.

38. The system of Claim 31, wherein the at least one processor instructs the display module to display the processed audit numbers as a piracy heat map.

39. A method for auditing a license for a software running on a network device, comprising:

- collecting machine parameters of the device, the collected machine parameters a combination of user-configurable and non-user-configurable machine parameters;

- generating a device identifier for the device based at least in part on the collected machine parameters;

- collecting a software identifier of the software running on the device;

- associating the software identifier with the device identifier to generate an audit number;

- transmitting the audit number to a server for storage in an audit database.

40. The method of Claim 39, further comprising:

- determining a geo-location code for the device; and

- associating the code with at least one of the software identifier and the device identifier to generate the audit number.

41. A method for auditing a license for a software running on a network device, comprising:

- collecting machine parameters of the device, the collected machine parameters a combination of user-configurable and non-user-configurable machine parameters;

- collecting at least one software identifier of the software running on the device;

- sending the machine parameters and the at least one software identifier to a server to generate a device identifier based at least in part on the machine parameters and the at least one software identifier;

- in response to receiving the device identifier, creating an audit number based at least in part on the received audit number; and

- transmitting the audit number to an audit database for storage.

42. The method of Claim 39, further comprising:

- determining a geo-location code associated with the device; and

- sending the code to the server to generate the device identifier based at least in part on the code.

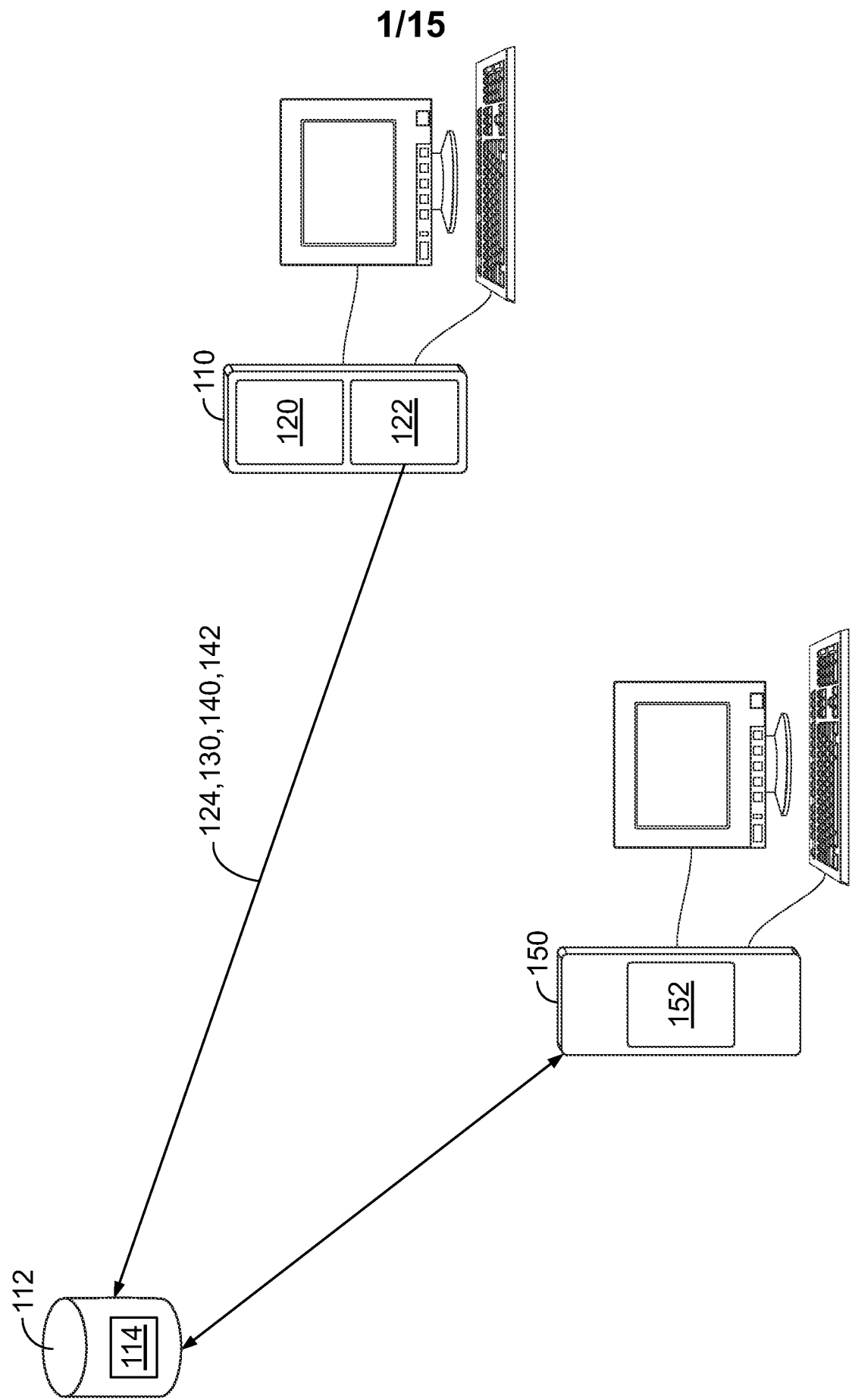


FIG. 1A

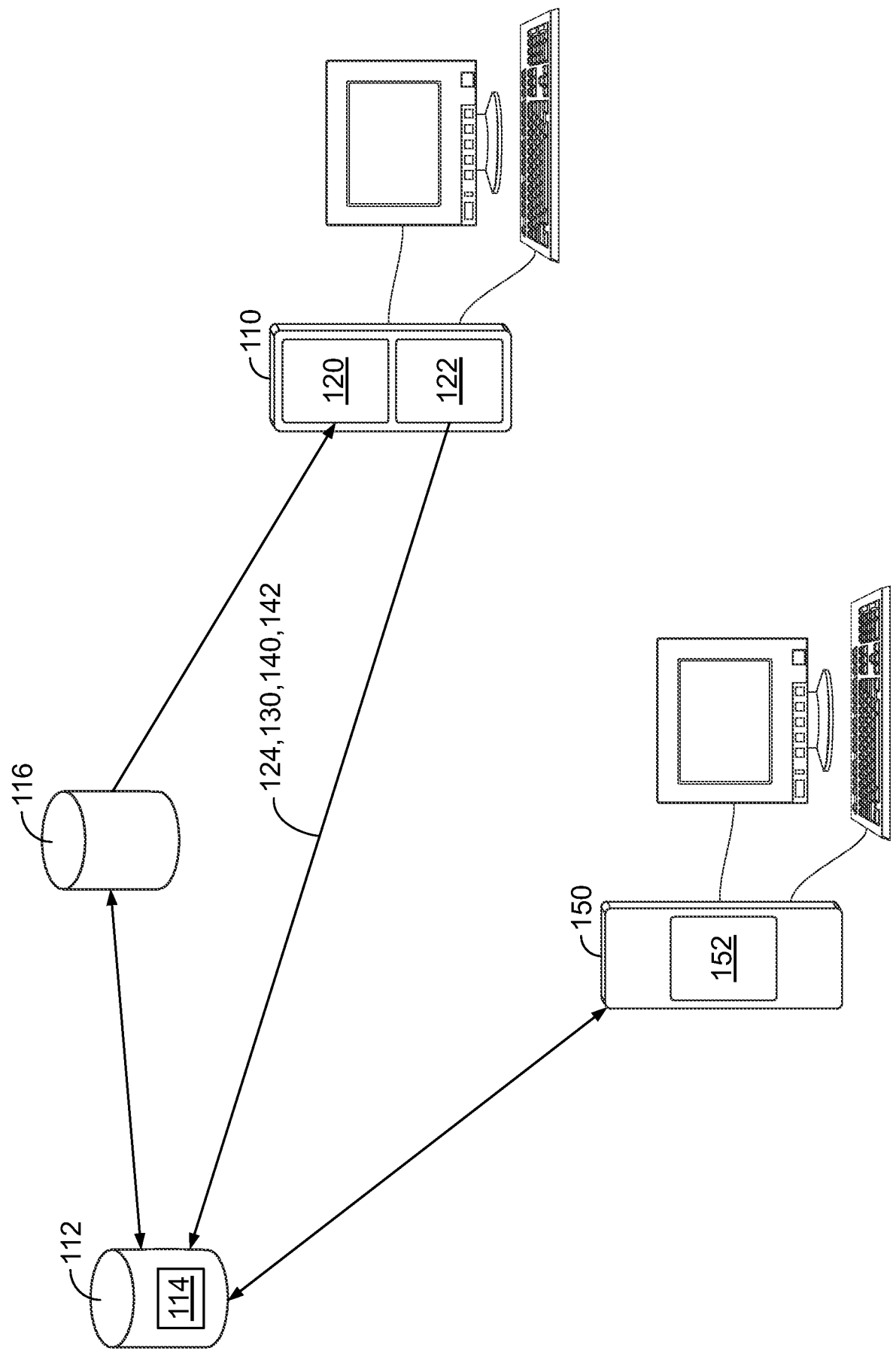


FIG. 1B

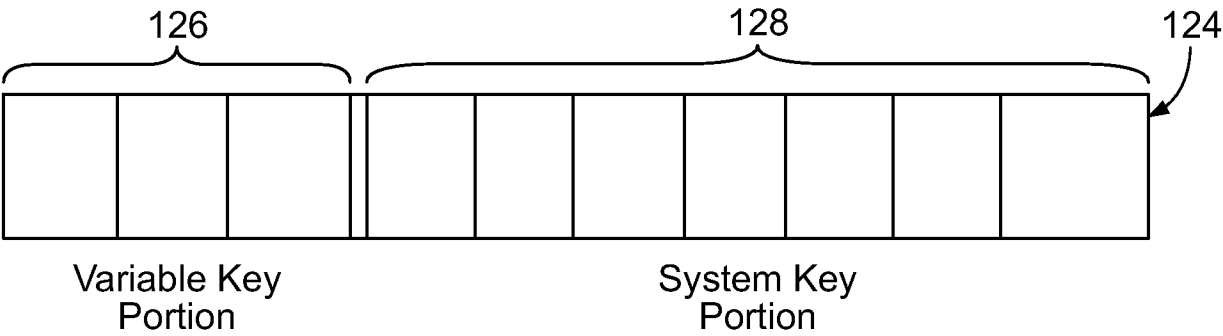


FIG. 2

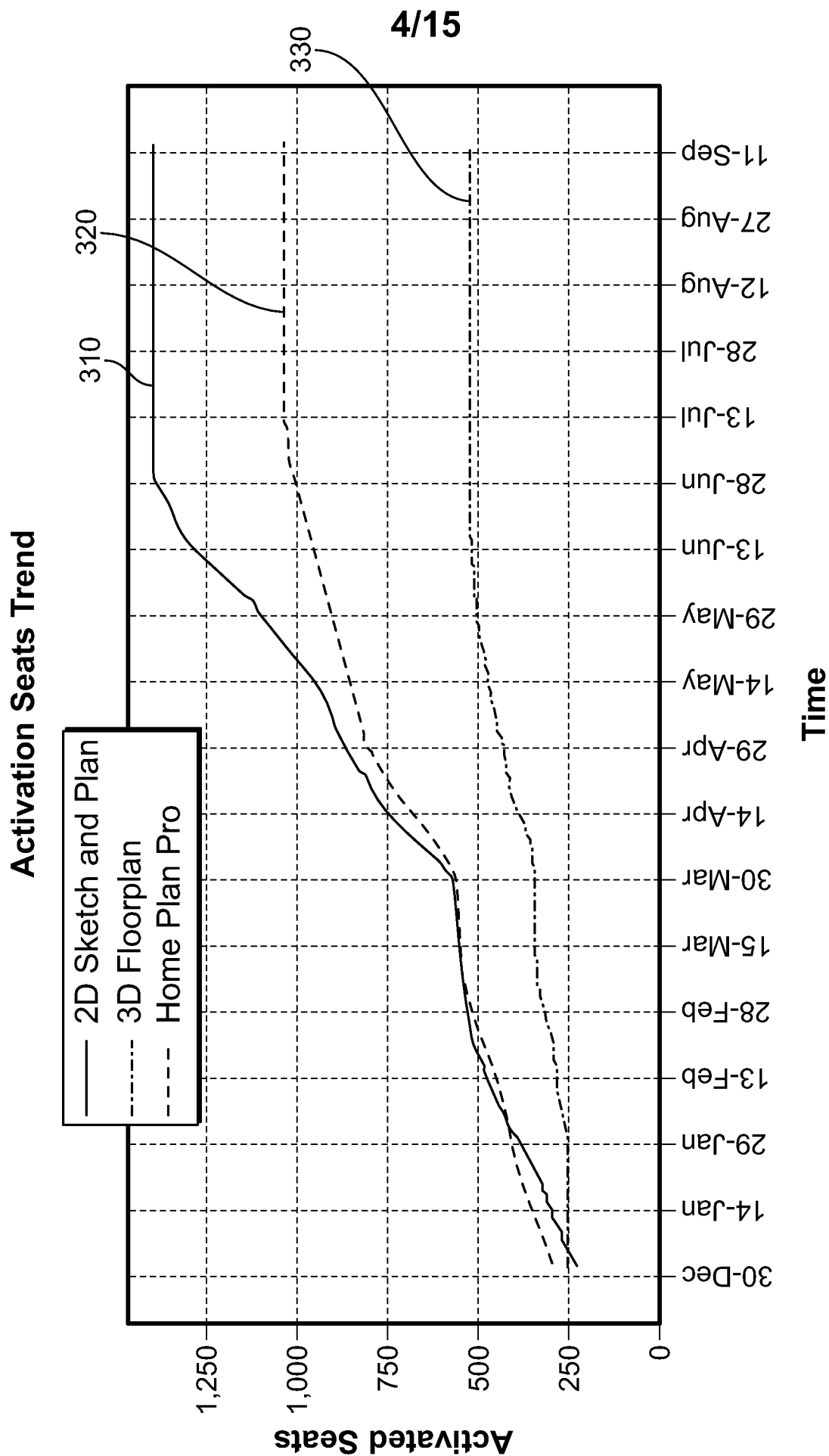


FIG. 3

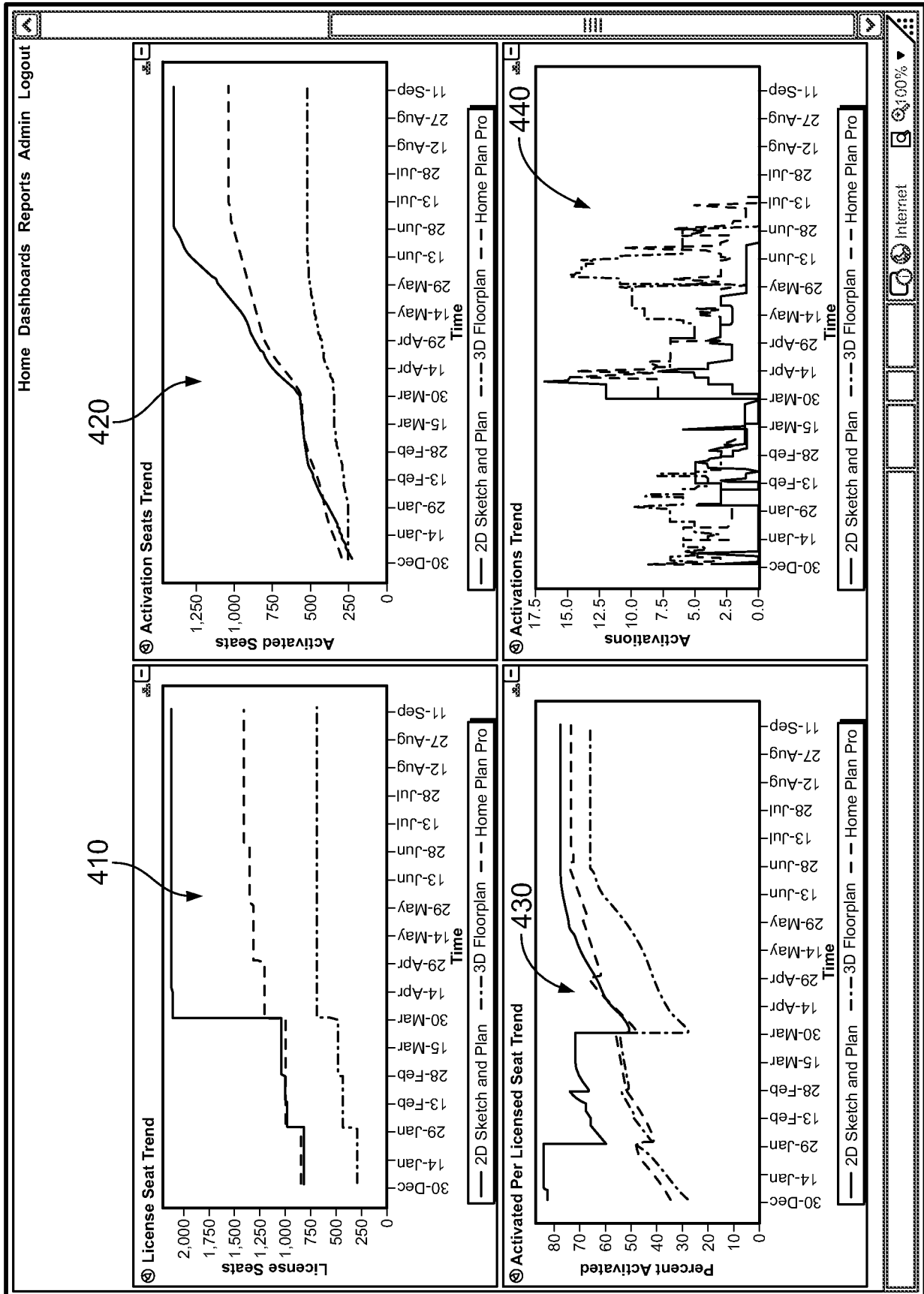
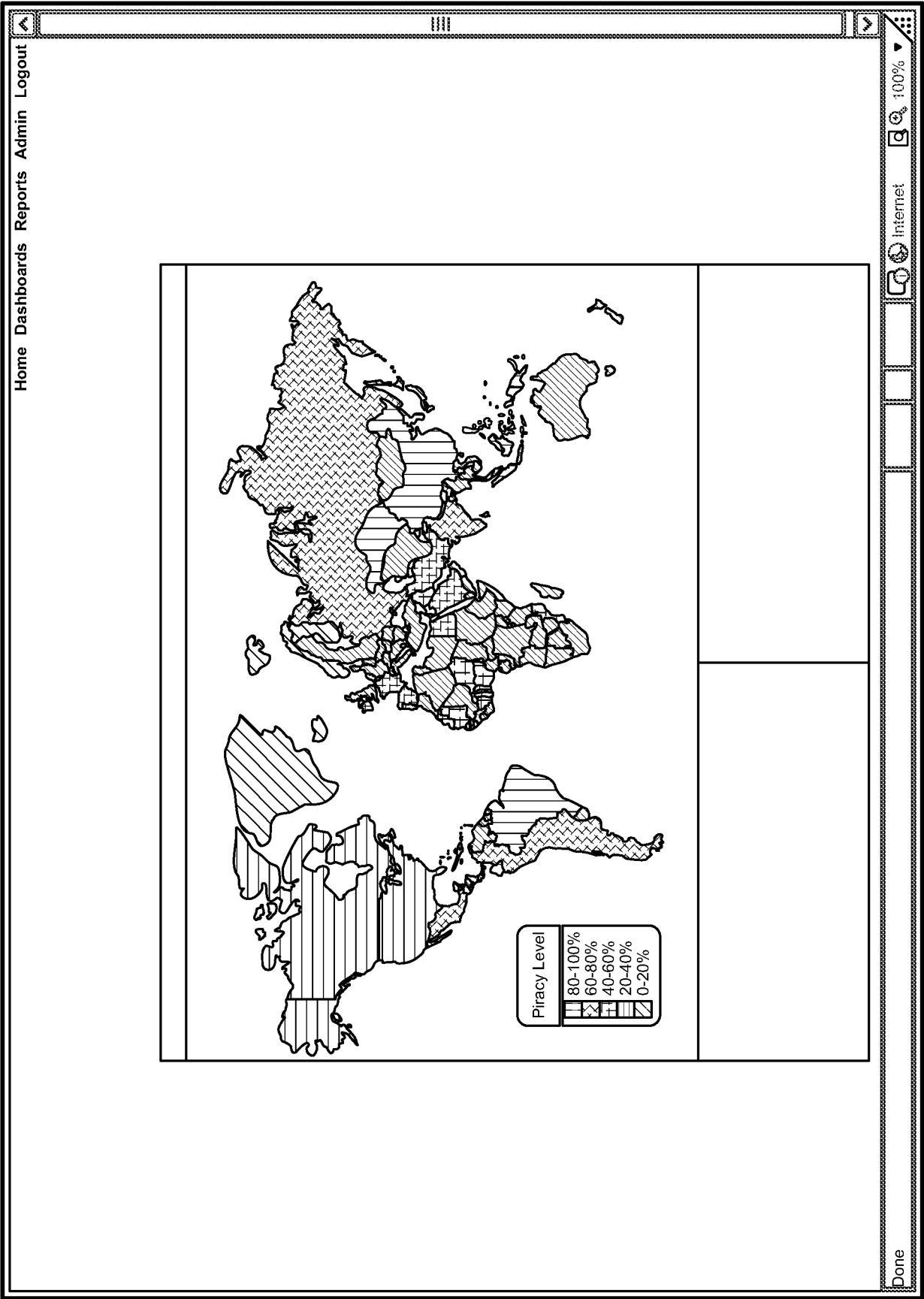
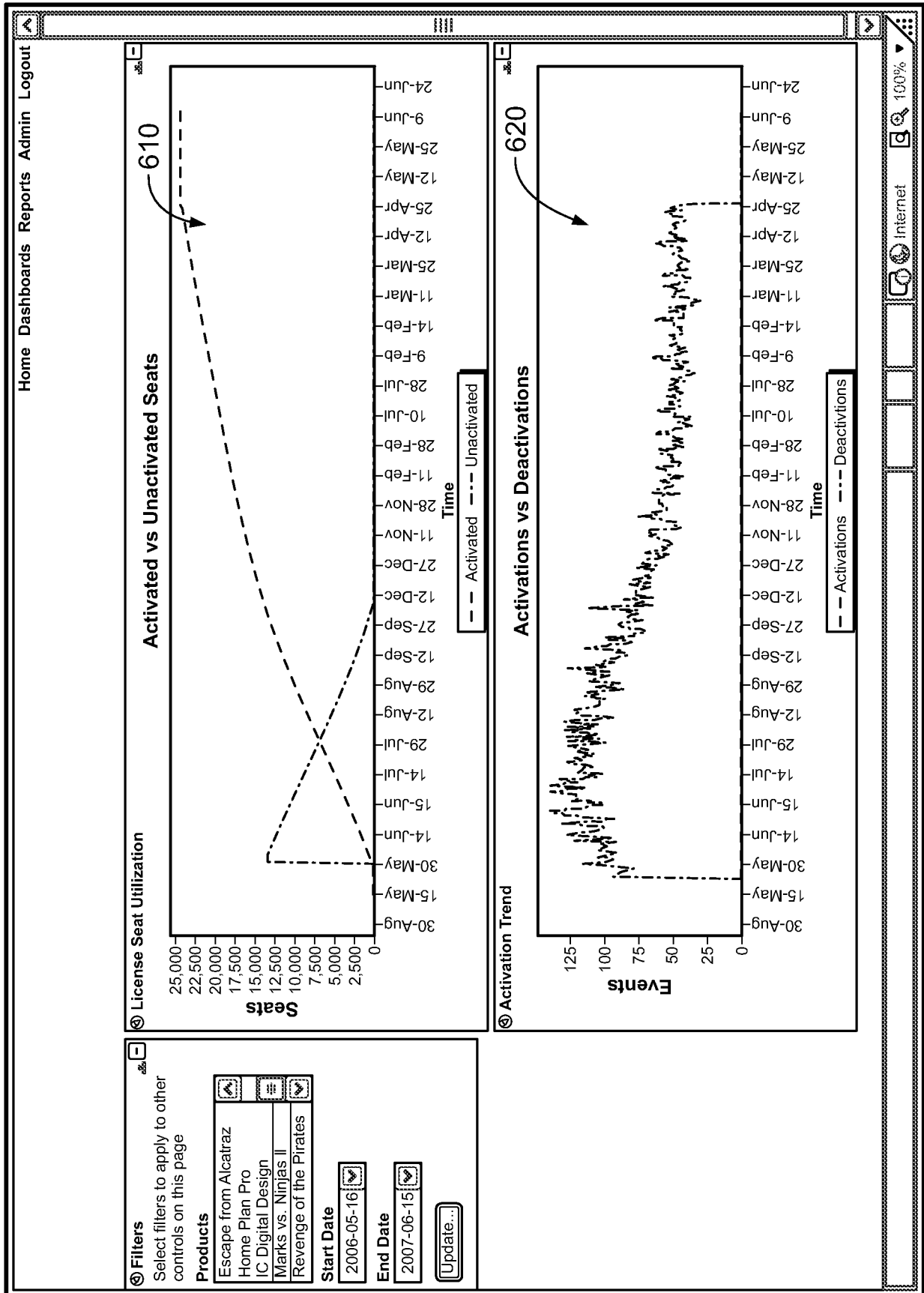


FIG. 5



7/15

FIG. 6



710

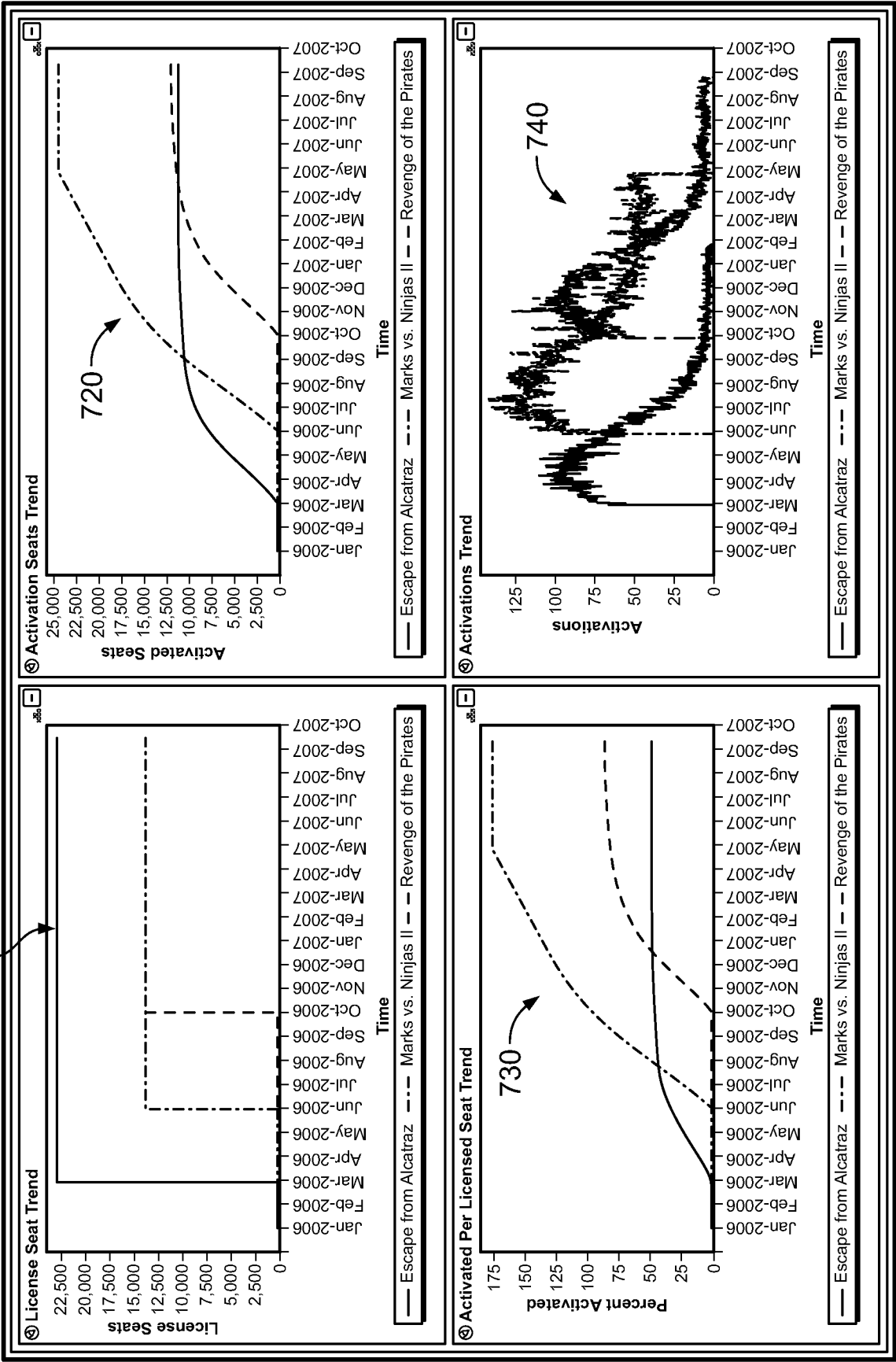


FIG. 7

9/15

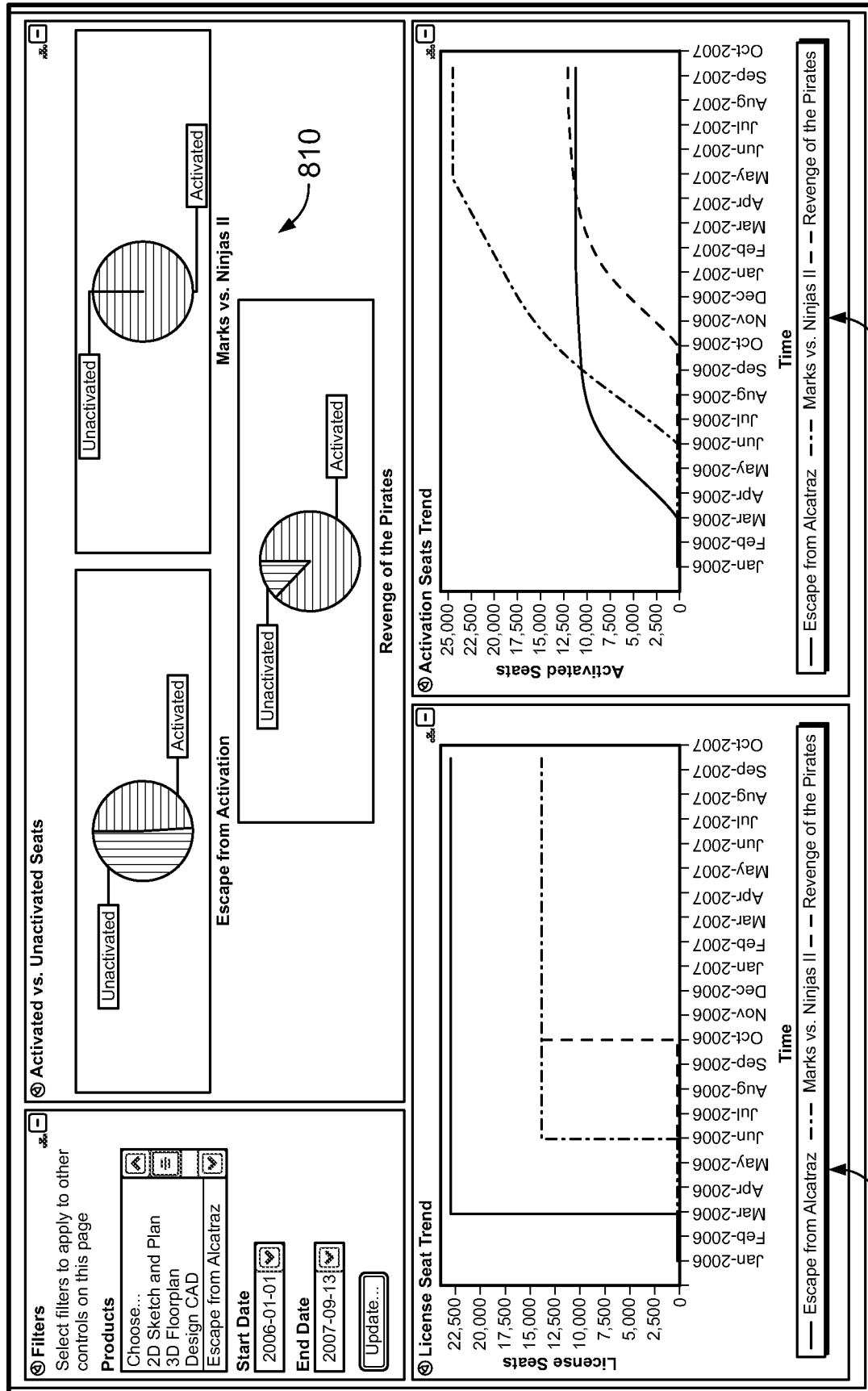
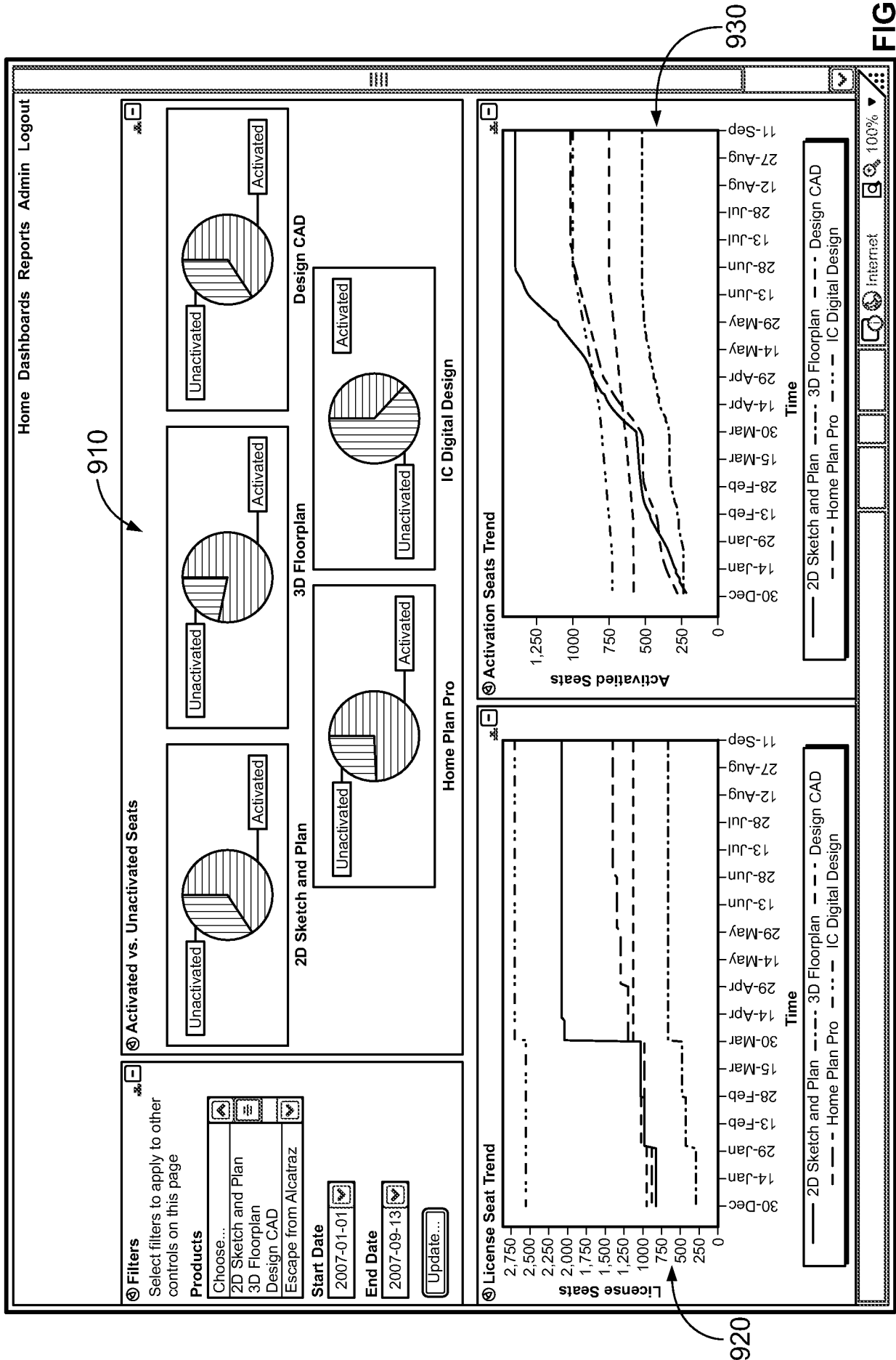


FIG. 8



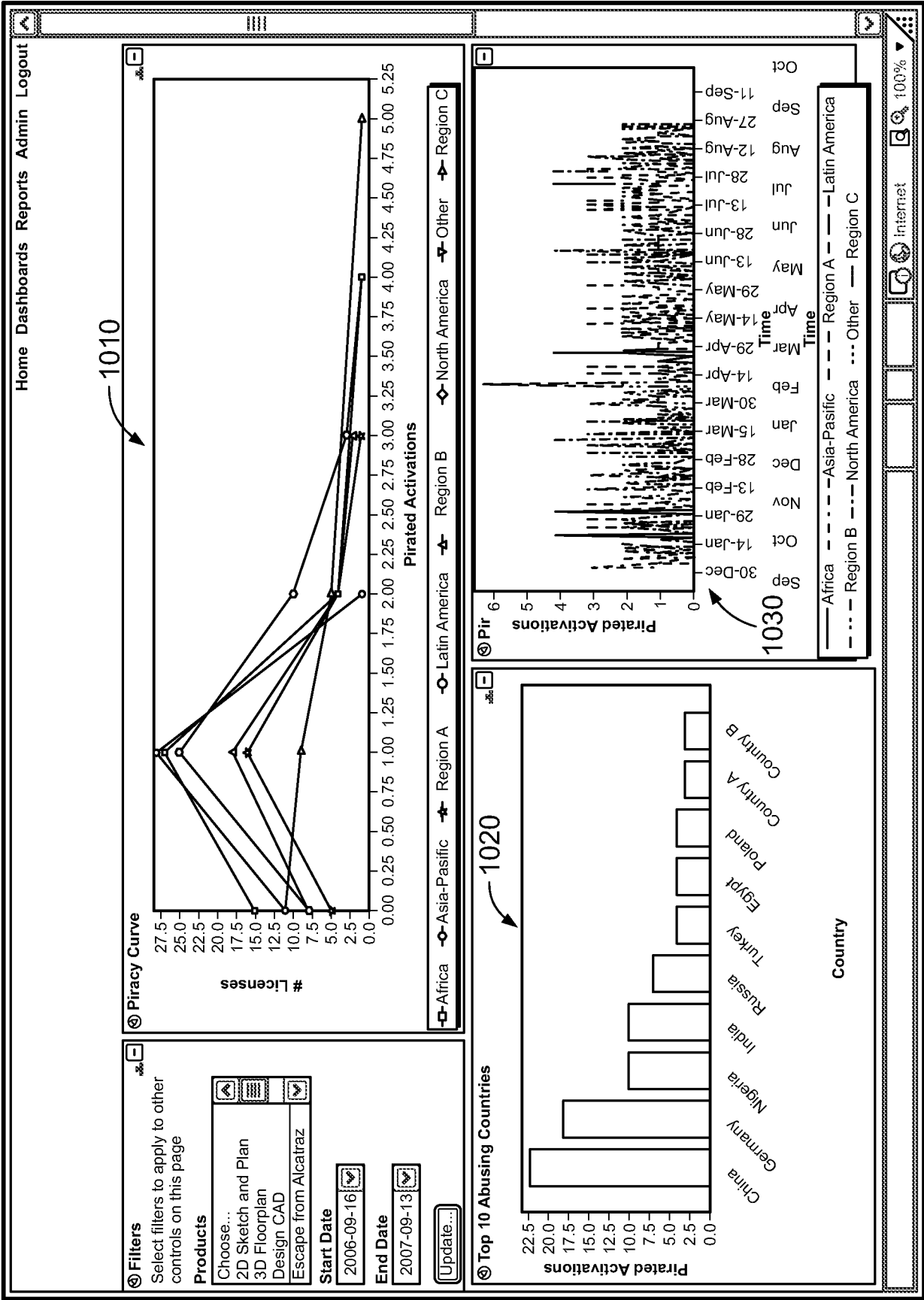
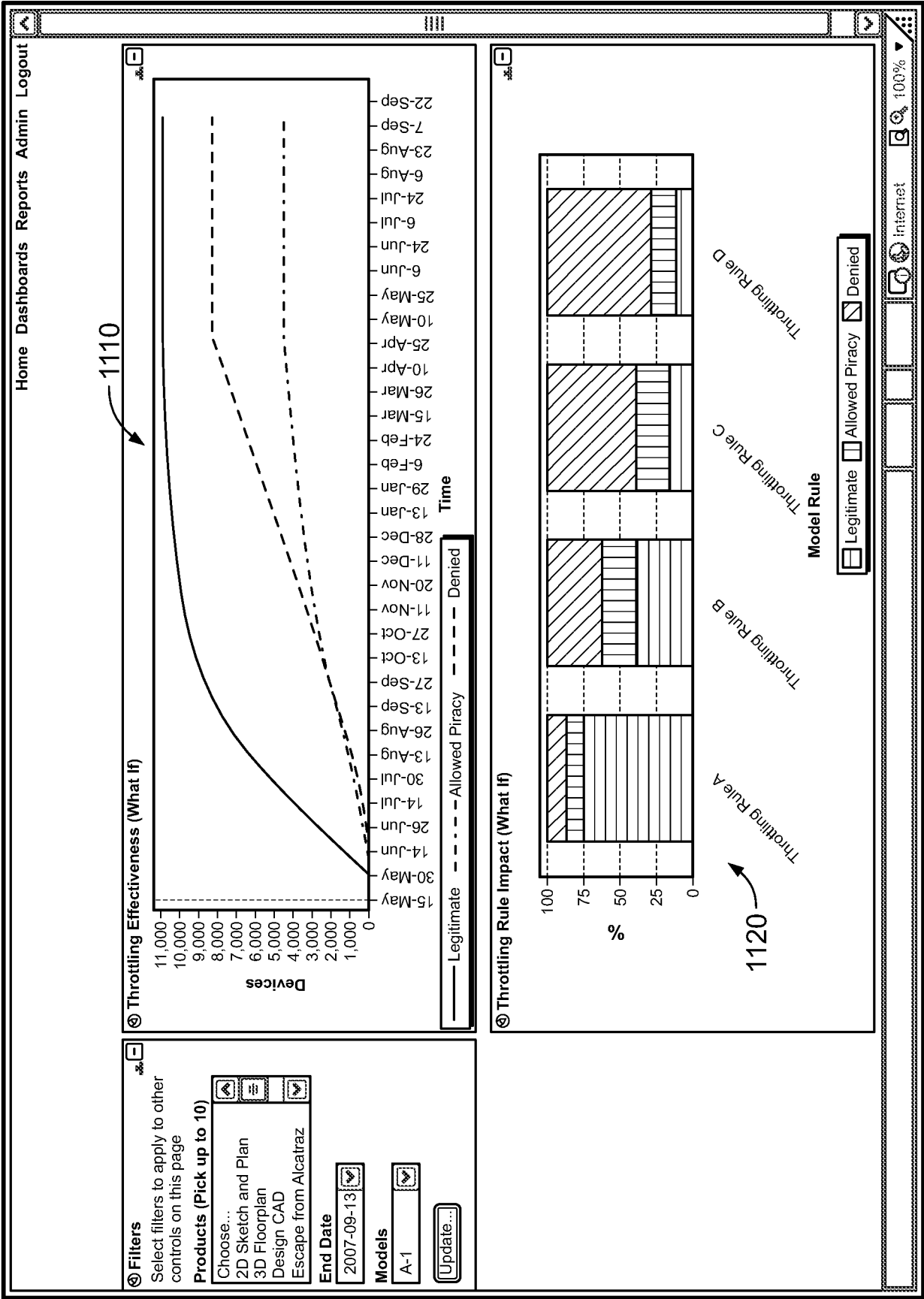
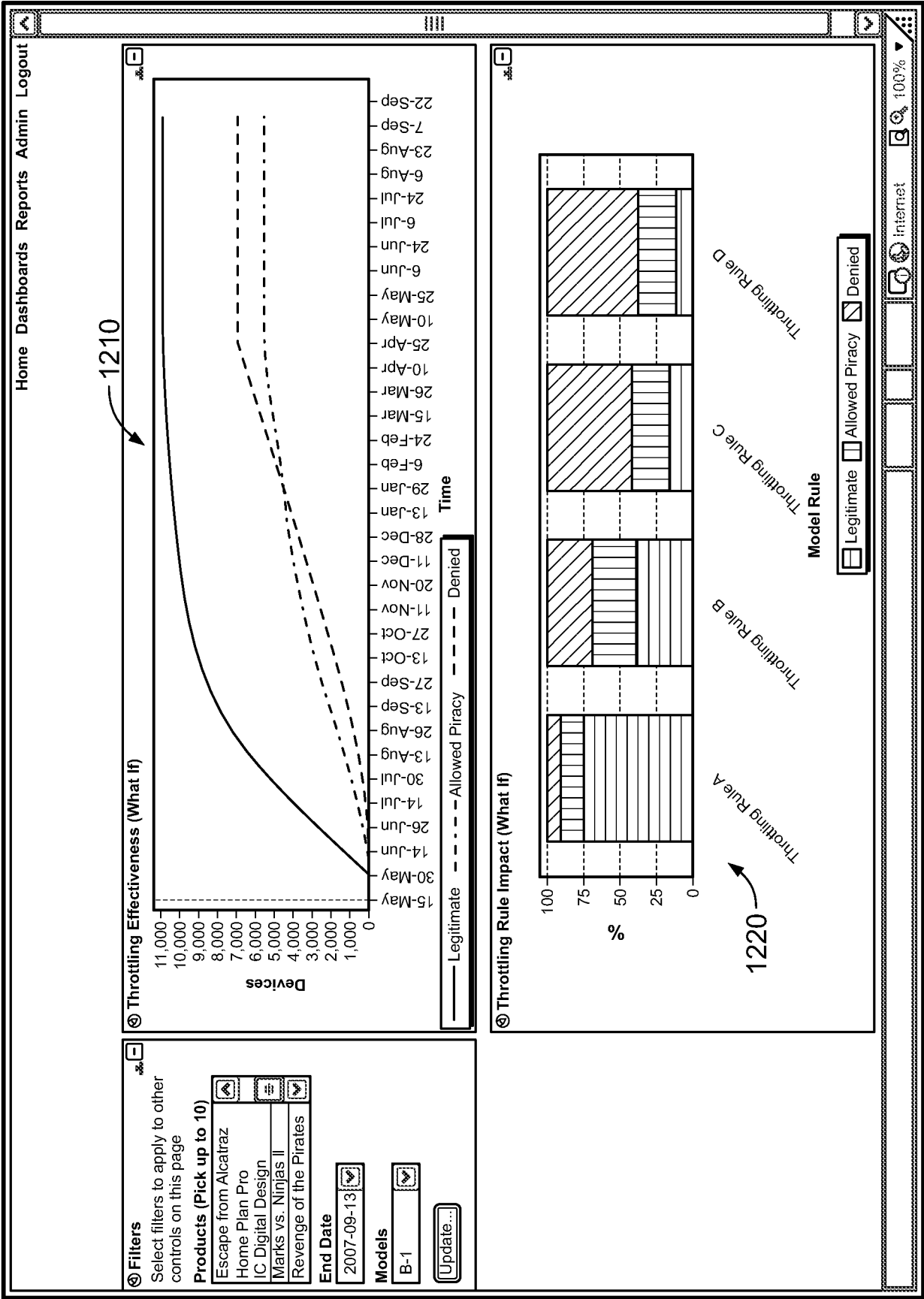


FIG. 10





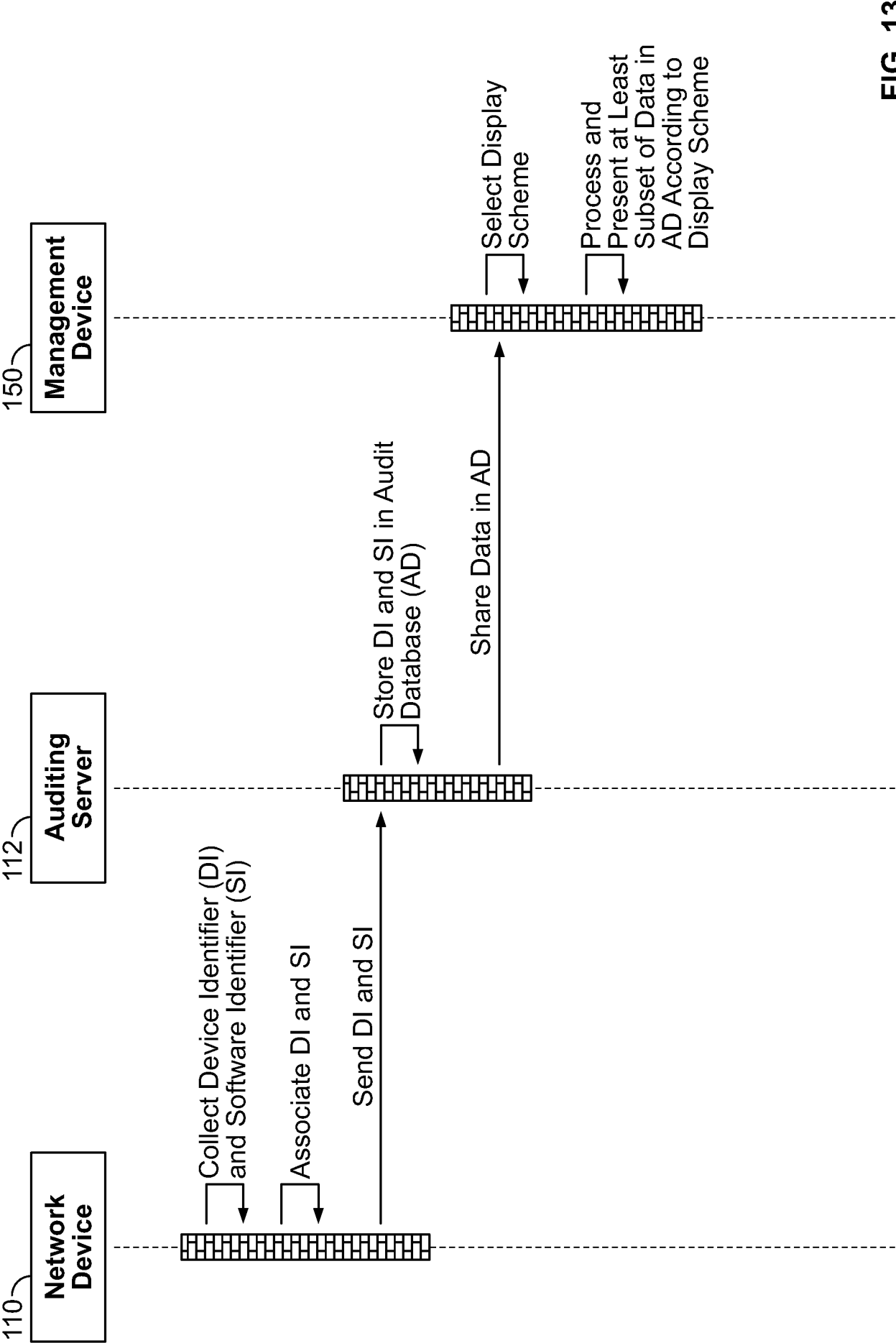


FIG. 13

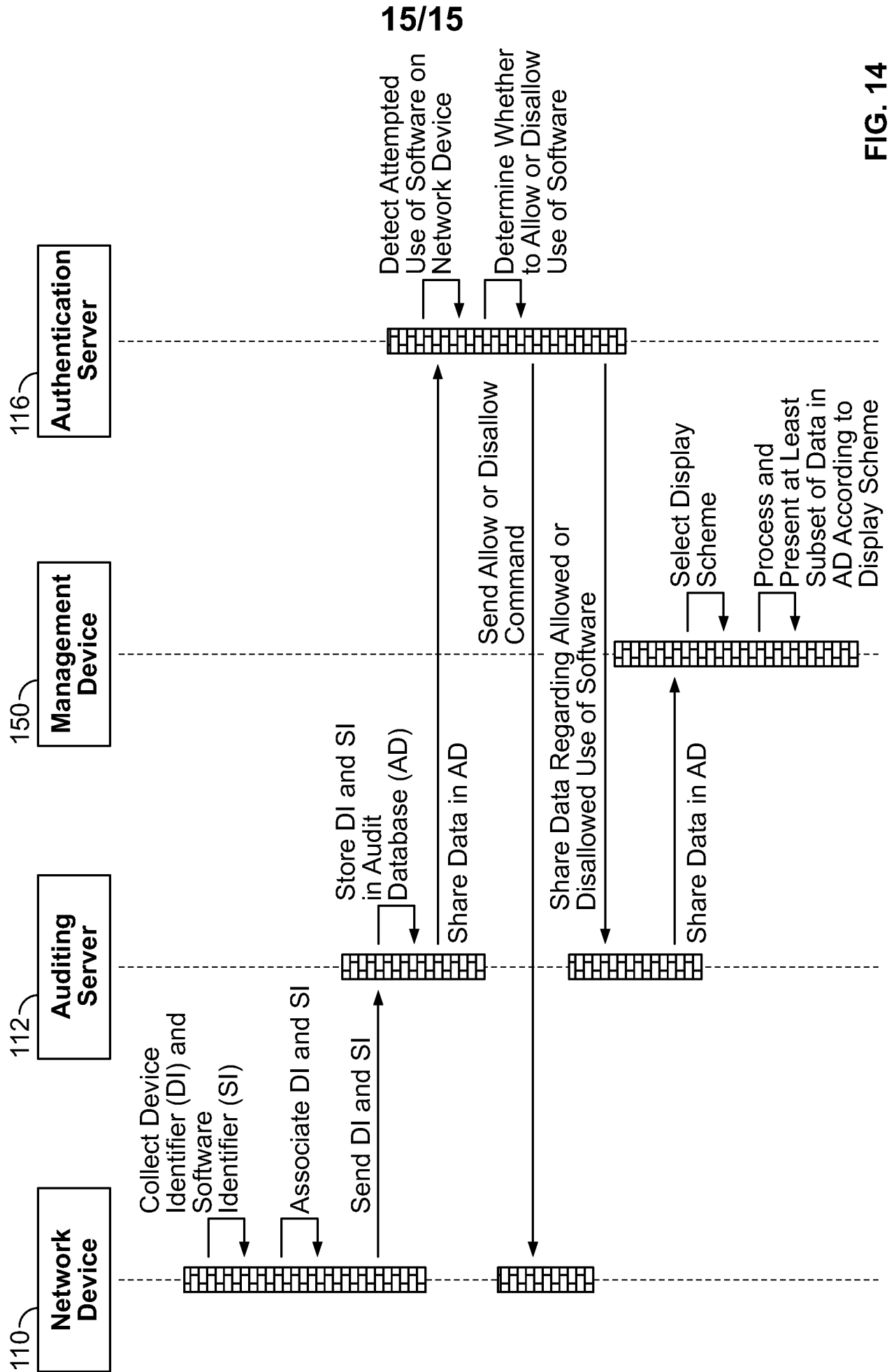


FIG. 14