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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 August 2016 (04.08.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/122697 A1

(51) International Patent Classification:
G06F 21/62 (2013.01) ***G06F 17/30*** (2006.01)

(21) International Application Number:
PCT/US2015/025768

(22) International Filing Date:
14 April 2015 (14.04.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

PCT/US2015/013987	30 January 2015 (30.01.2015)	US
PCT/US2015/019786	10 March 2015 (10.03.2015)	US
PCT/US2015/019788	10 March 2015 (10.03.2015)	US
PCT/US2015/019789	10 March 2015 (10.03.2015)	US
PCT/US2015/019792	10 March 2015 (10.03.2015)	US
PCT/US2015/019794	10 March 2015 (10.03.2015)	US

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) Inventor: **MILLER, Joseph A.**; 705 Taconic Trail, Petersburgh, New York 12138 (US).

(74) Agents: **FEBBO, Michael A.** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (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, 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, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

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

Published:

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

(54) Title: RESOURCE BROKERING FOR MULTIPLE USER DATA STORAGE AND SEPARATION

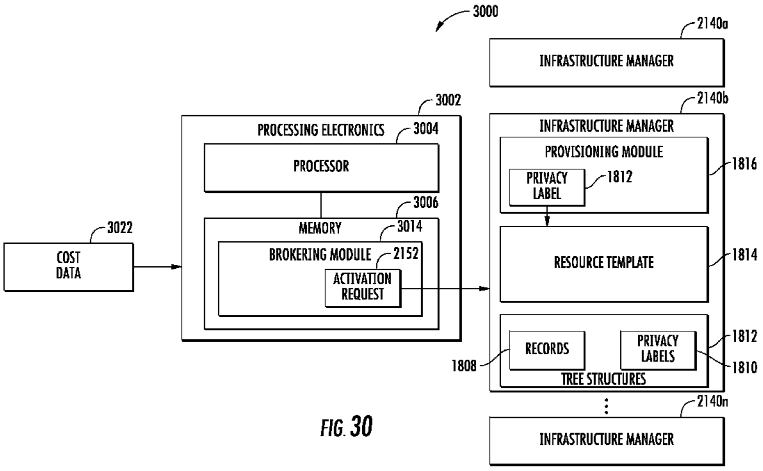

FIG. 30

(57) Abstract: A resource brokering apparatus includes memory and instructions stored in the memory. The instructions, when executed, cause a processor to receive cost data associated with hosting a resource from infrastructure managers, select one of the infrastructure managers to host the resource based on an analysis of the cost data, and send an activation request to the selected infrastructure manager requesting activation of a resource template to host the resource. The infrastructure manager includes an instruction to provision a privacy label to the resource template. The privacy label is correlatable with privacy labels stored in a database and defined by correlated tree structures having schema unrestricted by relational table structures. The privacy labels are also correlatable with records including user data items associated with multiple different users. The privacy labels distinguishing among the users.

RESOURCE BROKERING FOR MULTIPLE USER DATA STORAGE AND SEPARATION

BACKGROUND

[0001] Providers of cloud computing services may deploy multiple different business resources, such as email applications and case management systems. Each of these business resources may have multiple different clients. In some instances, client personnel may utilize shared business resources. In order to deploy multiple different business resources for multiple different clients, providers of cloud computing services typically deploy multiple servers with each server having its own operating system and multiple database instances. For example, a provider of cloud computing services may deploy 10 different business resources, with each business resource having 10 clients for a total of 100 different service instances. The service provider may deploy, for example, 10 different servers, with each server having an operating system and multiple database instances. Such databases are typically relational databases that enforce relational table structures, such as a column and row structure, to correlate and store data. As such, the service provider may deploy a separate relational database instance for each client on each of the different servers in order to provide appropriate data separation among clients.

[0002] Furthermore, user authentication and authorization for each dedicated server and database instance is typically implemented at the operating system level. As such, each user must be provisioned into the operating system in order to access the hosted business resource. Given this dependency on the operating system, the infrastructure and corresponding business resource instances are typically managed by the same service provider. Each newly deployed instance of a business resource typically requires a complex set of virtual machines to be configured in order to host the business resource. Given these interdependencies between infrastructure management and instance management, it remains challenging for providers of cloud computing services to take advantage of incremental improvements in infrastructure technology once a business resource has been deployed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Figure 1 is a schematic illustration of an example multiple user data storage and separation apparatus.

[0004] Figure 2 is a diagram illustrating example tree structures for facilitating the multiple user data storage and separation functionality of the apparatus of Figure 1.

[0005] Figure 3A is a Venn diagram illustrating an example relationship between receipts and labels defined by the tree structures of Figure 2.

[0006] Figure 3B is a Venn diagram illustrating an example relationship between receipts and nodes defined by the tree structures of Figure 2.

[0007] Figure 3C is a Venn diagram illustrating an example relationship among nodes, receipts, and labels defined by the tree structures of Figure 2.

[0008] Figure 4 is a diagram illustrating an example implementation of the tree structures of Figure 2.

[0009] Figure 5 is a flow diagram of an example process that may be carried out by the multiple user data storage and separation apparatus of Figure 1.

[00010] Figure 6 is a schematic illustration of an example authentication apparatus for use with the multiple user data storage and separation apparatus of Figure 1.

[00011] Figure 7 is a flow diagram of an example process that may be carried out by the authentication apparatus of Figure 6.

[00012] Figure 8 is a schematic illustration of another example authentication apparatus for use with the multiple user data storage and separation apparatus of Figure 1.

[00013] Figure 9 is a flow diagram of an example process that may be carried out by the authentication apparatus of Figure 8.

[00014] Figure 10 is a schematic illustration of an example authorization apparatus for use with the multiple user data storage and separation apparatus of Figure 1.

[00015] Figure 11 is a flow diagram of an example process that may be carried out by the authorization apparatus of Figure 10.

[00016] Figure 12 is a schematic illustration of another example authorization apparatus for use with the multiple user data storage and separation apparatus of Figure 1.

[00017] Figure 13 is a flow diagram of an example process that may be carried out by the authorization apparatus of Figure 12.

[00018] Figure 14 is a schematic illustration of an example data sandboxing apparatus for use with the multiple user data storage and separation apparatus of Figure 1.

[00019] Figure 15 is a flow diagram of an example process that may be carried out by the data sandboxing apparatus of Figure 14.

[00020] Figure 16 is a schematic illustration of another example data sandboxing apparatus for use with the multiple user data storage and separation apparatus of Figure 1.

[00021] Figure 17 is a flow diagram of an example process that may be carried out by the data sandboxing apparatus of Figure 16.

[00022] Figure 18 is a schematic illustration of an example resource provisioning apparatus for use with the multiple user data storage and separation apparatus of Figure 1.

[00023] Figure 19 is a flow diagram of an example process that may be carried out by the resource provisioning apparatus of Figure 18.

[00024] Figure 20 is a flow diagram of an example process that may be carried out by the resource template shown in Figure 18.

[00025] Figure 21 is a schematic illustration of another example resource provisioning apparatus for use with the multiple user data storage and separation apparatus of Figure 1.

[00026] Figure 22 is a flow diagram of an example process that may be carried out by the resource provisioning apparatus of Figure 21.

[00027] Figure 23 is a diagram of an example assignment of users, roles, and permissions to a tree of privacy labels.

[00028] Figure 24 is a schematic illustration of an example workflow management apparatus for use with the multiple user data storage and separation apparatus of Figure 1.

[00029] Figure 25 is a flow diagram of an example process that may be carried out by the workflow management apparatus of Figure 24.

[00030] Figure 26 is a schematic illustration of another example workflow management apparatus for use with the multiple user data storage and separation apparatus of Figure 1.

[00031] Figure 27 is a flow diagram of an example process that may be carried out by the workflow management apparatus of Figure 26.

[00032] Figure 28 is a schematic illustration of an example resource implementing workflow management for use with the multiple user data storage and separation apparatus of Figure 1.

[00033] Figure 29 is a flow diagram of an example process that may be carried out by the resource of Figure 28.

[00034] Figure 30 is a schematic illustration of an example resource brokering apparatus for use with the resource provisioning apparatus of Figures 18 and 21.

[00035] Figure 31 is a flow diagram of an example process that may be carried out by the resource brokering apparatus of Figure 30.

[00036] Figure 32 is a schematic illustration of another example resource brokering apparatus for use with the resource provisioning apparatus of Figures 18 and 21.

[00037] Figure 33 is a flow diagram of an example process that may be carried out by the resource brokering apparatus of Figure 32.

DETAILED DESCRIPTION OF EXAMPLES

[00038] Examples of resource brokering apparatus, systems, and methods for use with systems having multiple user data storage and separation functionality are disclosed herein. Multiple user data storage and separation functionality is of increasing interest to providers of cloud computing services desiring to provide multiple tenancy systems. In particular, the use of virtualization as opposed to server hardware duplication is increasing. Business resources, however, are currently developed from the ground up, often using available off-the-shelf applications, and have a typically large system footprint due in part to the limitations and complexities of relational database technologies used for data storage. These limitations and complexities may lead to, for example, the need for separate operating systems and database instances for multiple different business resources and system users in order to ensure appropriate data separation is maintained. This may in turn lead to high licensing and maintenance costs for deployment of business resources in multiple tenancy systems.

[00039] The limitations and complexities of relational database technologies may also negatively impact resource brokering in multiple tenancy systems. Given the need for separate operating systems and database instances for multiple different business resources and system users in order to ensure appropriate data separation is maintained, separating the management of infrastructure from the management of a business resource instance is impractical where data sets from multiple different entities and/or resources are required. Furthermore, complex sets of virtual machines must be configured in order to host each instance of a business resource. Given these complexities, automation of resource provisioning becomes time consuming and impractical to implement for a multiple tenancy system. Furthermore, given these interdependencies between infrastructure management and instance management, it becomes untenable for providers of cloud computing services to take advantage of incremental improvements in infrastructure technology once a business resource has been deployed. Cloud resources may have a sensitive business deployment model, particularly given the rapid advances in infrastructure technologies offering the advantages of, for example, greater efficiency, lower operational costs, and/or improved security. Because complex sets of virtual machines must be configured in order to host a new instance of the business resource, once a resource is deployed using a particular infrastructure, the incremental advantages of switching to a newer infrastructure are typically outweighed by the costs and complexities of redeploying the resource. Such complex redeployments are also disruptive to system users, who may experience delays associated with the redeployment and may be required to configure a new entry point for access to the resource.

[00040] The resource brokering apparatus, systems, and methods for use with systems having multiple user data storage and separation functionality disclosed herein may facilitate the implementation of a single centralized resource provisioning mechanism within a single computing system or virtualized operating system hosting parallel business resources and a single database instance that stores and separates data for multiple business resources and multiple users without resulting in bleeding or unauthorized disclosure of data from one user to another. In particular, the resource provisioning apparatus, systems, and methods for use with systems having multiple user data storage and separation functionality disclosed herein may utilize privacy labels distinguishing among multiple users. The privacy labels may be correlated with data records defined by correlated tree structures having schema such that the records are unrestricted by relational table structures, such as columns and rows. The use of such tree structures may impose fewer limitations on and provide greater flexibility of the structure of each record while at the same time maintaining appropriate data separation among users in a single database instance.

[00041] Incorporating privacy labels into the resource provisioning mechanism may provide for resource brokering to take advantage of advances in infrastructure technologies. By provisioning a privacy label to each newly deployed resource and appropriately provisioning permissions and privacy labels to users associated with the resource, separation of infrastructure management and instance management may be achieved while maintaining appropriate data separation. The centralized granular control of authentication and authorization for individual resources and data elements at the application level that is facilitated by the use of the privacy labels disclosed herein may remove the need to provision users at the operating system level. Resource instances may be created from resource templates and automatically deployed within a multiple tenancy system without the need to configure complex sets of virtual machines to provide infrastructure and access management for each newly deployed resource instance. In this way, a service provider may quickly and automatically deploy a multiple tenant system using a single service infrastructure and database instance. The service provider may also hand-off (e.g., to another provider such as an instance manager) or otherwise separate deployment and management of individual or groups of business resource instances within the system at the application layer such that new pre-defined or custom resource instances may be quickly and automatically deployed with appropriate constraints on system scale. The service provider may also be able to leverage advances infrastructure technologies by analyzing cost, security, efficiency, and other data for multiple different infrastructure providers, and automatically redeploying resources and/or transferring data to take advantage of lower costs and/or improved security and efficiency. Such redeployments and transfers may be transparent to system users, who may continue to access data and resources using the same point of entry provided by the service provider or a separate instance manager.

[00042] Figure 1 schematically illustrates an example multiple user data storage and separation apparatus 100. As will be described hereafter, apparatus 100 may include multiple user data storage and separation functionality. Apparatus 100 may be, for example, a component of a cloud computing system used to provide multiple different business resources to multiple different users 101. In particular, apparatus 100 may provide a single database instance that stores and separates data for multiple users 101 without resulting in bleeding or unauthorized disclosure of data from one user to another. The term "resource" as used herein refers to computer-based or network-based services in general, such as email applications, case management systems, etc. Resources may be, for example, resources used in capability-based systems having central access control and a data storage mechanism at the application layer rather than the operating system. While the examples set forth herein are primarily described in the context of "business" resources, it will be appreciated that other types of resources are contemplated as well.

[00043] Apparatus 100 may include processing electronics 102. Processing electronics 102 may include, for example, a processor 104 configured to execute logic in the form of instruction modules contained in a memory 106. For purposes of this application, the term "processor" shall mean a presently developed or future developed processor 104 that executes sequences of instructions contained in memory 106. In general, upon executing instructions contained in the memory, processor 104 may provide multiple user data storage and separation functionality in apparatus 100. The instructions may be loaded in a random access memory (RAM) for execution by the processor 102 from a read only memory (ROM), a mass storage device, or some other persistent storage. In some examples, hardwired circuitry modules may be used in processing electronics 102 in place of, or in combination with, processor 104 and/or instruction modules stored in memory 106 to implement the multiple user data storage and separation functionality described herein. For example, the multiple user data storage and separation functionality of apparatus 100 may be implemented entirely or in part by logic contained in an application-specific integrated circuit (ASIC). Unless otherwise specifically noted, processing electronics 102 is not limited to any specific combination of hardware circuitry modules and instruction modules, nor to any particular source for instructions executed by processor 104.

[00044] Memory 106 may include a non-transitory computer-readable medium. The term “non-transitory computer-readable medium” as used herein includes any computer readable medium, excluding only transitory propagating signals per se. Memory 104 may include, for example any non-volatile or volatile memory such as DRAM, RAM, ROM, register memory, or some combination of these; for example a hard disk combined with RAM. Memory 106 may store software instruction modules for execution by processor 104. In some examples, memory 106 may further store data for use by processor 104. Memory 106 may store various software instruction modules that direct processor 104 to carry out various interrelated actions, such as the multiple user data storage and separation functionality of apparatus 100.

[00045] As shown in Figure 1, memory 106 may include records 108 stored therein. Records 108 may be, for example, individual user data items or groupings of associated user data items. For example, apparatus 100 may be a component of a cloud computing system that provides an email application, and memory 106 may store email records. Such records 108 may be comprised of associated user data items such as “Subject” data from the subject lines of emails, “Message” data including the text of emails, “To” data including the identity of message recipients, and “From” data including the identity of message senders. Memory 106 may store data items for multiple different users 101. The term “user” as used herein may refer to single individuals (e.g., clients or employees) or entities (e.g., companies or corporate entities) or grouping or subgroupings of individuals and/or entities (e.g., an employer entity, an IT department subgroup entity, and subgroups of individual employees), as well as computer devices and/or systems (e.g., an autonomous computer system capable of interaction with apparatus 100).

[00046] Records 108 may be defined by and stored using correlated tree structures 110. For example, records 108 may be defined by node-based binary tree structures used for data storage and searching, such as self-balancing binary tree structures. In particular, tree structures 110 may have schema definitions such that stored records 108 are unrestricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns. In some examples, the structures of two different records 108 may not be the same. For example, continuing with the email example from above, some records 108 may be structured and stored by associating “Subject”, “Message”, “To”, and “From” user data items, while some records 108 may be structured and stored by associating only “Subject”, “Message”, “To” user data items. In this example, the schema defined by tree structures 110 may not require null or placeholder “From” data for those records 108 that are structured to exclude it, as opposed to a relational table structure, which would require null “From” values in each record 108.

[00047] Memory 106 may also include privacy labels 112. Privacy labels 112 may distinguish among multiple different users 101. Each privacy label 112 may include a unique identification mapping to a user 101. In some examples, privacy label 112 may include a number of tokens (e.g., unique randomly generated cryptographically entropic values having a predetermined number of bytes, etc.) that are uniquely provisioned to user 101. For example, multiple different users 101 may include a manager who shares a subset of records 108 with an employee. The manager may be provisioned a privacy label 112 having a particular set of N tokens. The employee may be provisioned with a privacy label 112 having, for example, a subset of the N tokens.

[00048] Privacy labels 112 may be correlated with records 108 whenever user data items associated as records 108 are written to memory 106. For example, the manager’s privacy label 112 including the full set of N tokens may be correlated with all records 108 written to memory 106 by the manager. A different privacy label 112 containing a subset of the N tokens may be correlated with all records 108 written to memory 106 by the employee.

[00049] Once privacy labels 112 are correlated with records 108, they may be compared with user identifier data provided by users 101 requesting records 108 from memory 106 in order to ensure that only records 108 associated with a particular user 101 are accessed by that user 101. For example, the manager may provide user identifier data including the full set of N tokens along with query data to apparatus 100. The full set of N tokens in the user identifier data may be compared with privacy labels 112 correlated with records 108 in order to ensure that only records 108 with correlated privacy labels 112 including the full set of N tokens provisioned to the manager, and records 108 with correlated privacy labels 112 including the subset of N tokens provisioned to the employee are returned to the manager. Even if the user identifier data provided by the manager includes the full set of N tokens, the manager may still access records 108 having fewer than N correlated tokens, provided that all tokens for a record 108 are included in the full set of N tokens provided by the manager. Similarly, the employee may provide user identifier data including the subset of N tokens along with query data to apparatus 100. The subset of N tokens in the user identifier data may be compared with privacy labels 112 in order to ensure that only records 108 correlated with the subset of N tokens provisioned to the employee are returned to the employee. The employee may not access records 108 for which there are a greater number of tokens than those provided by the employee.

[00050] Other provisioning schemes for privacy labels 112 are contemplated as well. For example, unique privacy labels 112 having different combinations of a number of bytes may be provisioned to each user rather than provisioning subsets of tokens from a full set of tokens. As such, multiple different users may include users for multiple different business resources, multiple different entities, etc. Records 608 associated with each user stored in a single database instance. As will be understood, records 108 for each of multiple different users 101 may be correlated with privacy labels 112 in order to ensure that only records 108 associated with a particular user 101 are accessed by that user 101. A user 101 may be associated or disassociated with a particular record 608 by correlating or removing a privacy label 112, as opposed to copying or replicating individual records 608 into multiple different database instances for different resources or clients.

[00051] Privacy labels 112 may be correlated with each of records 108 using privacy label tree structures included in tree structures 110. The privacy label tree structures defining privacy labels 112 may have schema definitions similar to other tree structures 110. That is, privacy labels 112 may be structured and stored without being restricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns.

[00052] By way of example, Figure 2 is a diagram illustrating example tree structures 200 for facilitating the multiple user data storage and separation functionality of apparatus 100. As shown in Figure 2, the tree structures 200 defining records may include segment trees 202 and receipt trees 204. Segment trees 202 may include a number of correlated trees corresponding to segments 206. Segments 206 may correspond to, for example, a particular type or grouping of user data items. For example, as shown in Figure 2, segments 206 may include a segment 206a for "Subject" data from the subject lines of emails, a segment 206b for "Message" data including the text of emails, a segment 206c for "To" data including the identity of message recipients, and a segment 206d for "From" data including the identity of message senders. Each tree structure corresponding to a segment 206 may include a number of nodes 208. Each node 208 may in turn be an individual node, may be a parent node having a number of child nodes, or may be a child node depending from a parent node. Each node 208 may contain a user data item or a reference to a user data item (e.g., if a user data item is stored in a location other than memory 106). For example, node 208a may include particular "Subject" data for an email, node 208b may include particular "Message" data for an email, node 208c may include particular "To" data for an email, and node 208d may include particular "From" data for an email. Each node 208 may also contain a reference to a number N of receipts 210. That is, each of the nodes 208 in a segment tree 206 has a corresponding receipt tree 204 to define a respective set of records 108 stored in memory 106 in which that respective node 208 is included.

[00053] Receipt trees 204 may include a number of correlated trees corresponding to receipts 206. Each receipt 206 may correspond to and define a respective set of N associated nodes 208, where each of the nodes 208 contains a reference to the respective receipt 210. Accordingly, each node 208 may list a number N of receipts 210, and each receipt 210 may list an unrelated number N of nodes 208. The set of N associated nodes for a particular receipt 210 may define a record 108. For example, each of nodes 208a, 208b, 208c, and 208d may contain a reference to a receipt 210a. Figure 2 illustrates an example reference to a receipt 210a for node 208c. In turn, receipt 210a may contain references to each of nodes 208a, 208b, 208c, and 208d. Figure 2 illustrates an example reference to node 208c for receipt 210a. As such, receipt 210a may define a record that includes each of nodes 208a, 208b, 208c, and 208d. It will be appreciated that the additional receipt references for node 208c shown in Figure 2 may represent the inclusion of node 208c in N-1 other records 108 that may contain additional, fewer, or different nodes 208 than those for record 108a defined by receipt 210a.

[00054] As shown in Figure 2, tree structures 200 may also include privacy label trees 212. Privacy label trees 212 may include a number of correlated trees corresponding to privacy labels 214. Privacy labels 214 are correlated with receipts 210 in receipt trees 204 to define respective sets of records 208 associated with respective different users 101. In particular, each privacy label 214 may correspond to and define a respective set of N associated receipts 210, where each of the receipts 210 contains a reference to the respective privacy label 214. Accordingly, each privacy label 214 may list a number N of receipts 210, and each receipt 210 may list an unrelated number N of privacy labels 214. In some examples, for each privacy label 214, there may or may not be a corresponding receipt 210, but for each receipt 210, there must be at least one privacy label 214. The set of N associated receipts 210 for a particular privacy label 214 may define a set of records 108 for which a particular user 101 may gain access, provided that particular privacy label 214 has been provisioned to that particular user 101.

[00055] As will be appreciated, each node 208 may have a number N of corresponding receipts 210, which may, in turn, have a number N of corresponding privacy labels 214, which may provide a theoretically unbounded storage capability, limited only by the details of the particular practical implementation. As may also be appreciated, segments 206 and their corresponding nodes 208 may be grouped according to any particular attribute and are not limited in structure by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns. As such, segments 206 and their corresponding nodes 208 may be easily repurposed and/or reused simply by addition, deletion, or modification of corresponding receipts 210.

[00056] Figures 3A, 3B, and 3C are Venn diagrams providing an alternative representation of the relationships among tree structures 200 shown in Figure 2. In Figures 3A, 3B, and 3C, set L is the set of all receipts for all privacy labels, set R is the set of all receipts, and set K is the set of all receipts for a given node in a segment tree. Each privacy label may have many corresponding receipts. Each receipt may have many corresponding nodes. Each node may have a corresponding receipt tree of receipts. For purposes of Figures 3A, 3B, and 3C, it may be assumed that, for each privacy label, there may or may not be a corresponding receipt, but for each receipt, there must be at least one privacy label. Under this assumption, while all members of set R must correspond to one or more members (i.e., lists of corresponding receipts) of set L, not all members in set L will correspond to elements of set R. $L \cap R$ must always equal R, and therefore $R \subseteq L$ as shown in Figure 3A.

[00057] Referring now to Figures 3B and 3C, set S is the set of all nodes having an associated binary receipt tree, and S_0, S_1, … S_n are all binary trees. Assuming each receipt tree has multiple corresponding nodes, for each $x \in S_n$, K is the set of receipts for each node x. $\forall x \in S_n$: $\forall y \in K \cap R$ where K is not {} and K is the set of receipts for node x as shown in Fig. 3B. Thus, each node is directly correlated with a privacy label upon assignment of a receipt. Each receipt may reference up to S_n nodes. Each privacy label may reference up to N receipts. Accordingly, the total set of all nodes corresponding to a given privacy label is the intersection of nodes in the sets of S, for nodes corresponding to a receipt in K, by which there is a corresponding privacy label set in R, as shown in Figure 3C.

[00058] Figure 4 is a diagram illustrating an example implementation 400 of tree structures 200. Implementation 400 includes correlated binary tree structures corresponding to segments 402, receipts 404, and privacy labels 406. The three possible segments are segment 402a, corresponding to “Name” data, segment 402b, corresponding to “Age”, and a segment 402c, corresponding to “Favorite Day” data. Figure 4 illustrates three specific examples. In the first example, a record that uses only one of three possible segments 402 is illustrated. In particular, segment 402a is used, and a node 408a is assigned to Name data “Bob”. Node 408a has a corresponding receipt 410a. In the second example, a record that uses two of three possible segments 402 is illustrated. In particular, segments 402a and 402b are used, and nodes 408b and 408c are assigned to Name data “Joe” and Age data “21” respectively. Nodes 408b and 408c both correspond to receipt 410b. In the third example, a record that uses all three possible segments 402 is illustrated. In particular, segments 402a, 402b, and 402c are used. Node 402d is assigned to Name data “Ted”. Node 402e is assigned to Favorite Day data “Tuesday”. Node 408c is reused. Nodes 402c, 402d, and 402e correspond to receipt 410c. The records for Bob and Ted share the same privacy label 412a (e.g., the records for Bob and Ted were created by the same user), while the record for Joe is correlated with privacy label 412b.

[00059] Referring again to Figure 1, memory 106 may include receiving module 114, label identification module 116, query processing module 118, and privacy module 120. Modules 114, 116, 118 and 120 may cooperate to cause processing electronics 102 to carry out the process 500 set forth by the flow diagram of Figure 5. As indicated by a step 502, receiving module 114 may receive a communication 121 including user identifier data 122 and query data 124. For example, each privacy label 112 may include a unique identification mapping to a user 101. In some examples, privacy label 112 may include a number of tokens that are uniquely provisioned to user 101. User 101 may send a communication 121 including user identifier data 122 that includes a set of N tokens along with a query to apparatus 100.

[00060] In some examples, user identifier data 122 is out of band data with respect to query data 124. That is, user identification data 122 may be received by receiving module 114 in the same communication as query data 124, but may be kept separate from query data 124 by a conceptually independent data channel provided as an inherent characteristic of the communication channel and transmission protocol, as opposed to requiring a separate communication channel and endpoints to be established at apparatus 100. In this way, user 101 need not be aware of and is not required to enter user identification data 122, and user identifier 124 may be processed independent of query data 124 by label identification module 116 and privacy module 120. Query data 124 may be, for example, a select statement or other type of query that defines the scope of a data request. For example, query data 124 may include a request to retrieve all emails from person X having Y in the subject line. Other types of query data 124 are contemplated as well.

[00061] At a step 504, label identification module 116 may identify a set of privacy labels 112 based on user identifier data 122. In some examples, label identification module 116 identifies the set of the privacy labels 112 by comparing tokens in the user identifier data 122 with tokens in privacy labels 112 stored in memory 106. In some examples, if tokens in the user identifier data 122 do not match tokens in any privacy labels 112, user 101 is informed that no data exists. In some examples, label identification module 116 communicates with a privacy label mapping service to obtain the actual tokens for the ones the user is providing (e.g., user 101 is not provided with the actual tokens used by apparatus 100, but rather reference tokens).

[00062] At a step 506, query processing module 118 may identify a set of records 108 based on query data 124. In some examples, query processing module 118 may identify the set of records 108 using tree structures such as those described with reference to Figure 2. For example, query processing module 118 may identify segment tree nodes associated with query data 124, and further identify receipts correlated with the segment tree nodes associated with the query data using receipt trees correlated with the segment trees. By way of example, query data 124 may include a request to retrieve all emails from person X having Y in the subject line. Query processing module 118 may search segment trees for segments corresponding to “From” and “Subject” in order to identify nodes matching X and Y. Query processing module 118 may then identify receipts correlated with nodes matching X and Y. The identified set of receipts may correspond to a set of records 108 meeting the requirements of query data 124.

[00063] At a step 508, privacy module 120 may return, in response to communication 121, only records 126 from the set of records 108 identified by query processing module 118 that are associated with the set of the privacy labels 112 identified by label identification module 116. In some examples, privacy module 120 may identify the appropriate records 108 using tree structures such as those described with reference to Figure 2. For example, privacy module 120 may identify privacy labels 112 correlated with the receipts that correspond to a set of records 108 meeting the requirements of query data 124. Privacy module 120 may then compare the resulting set of privacy labels 112 with the set of the privacy labels 112 identified by label identification module 116. Privacy module 120 may then return only those records 108 for which the set of privacy labels 112 identified by privacy module 120 match those identified by label identification module 116. If there are no matching privacy labels 112, then privacy module 120 may respond to the communication indicating no data exists.

[00064] Figure 6 schematically illustrates an example authentication apparatus 600 for use with a multiple user data storage and separation apparatus, such as apparatus 100 shown in Figure 1. Apparatus 600 may be, for example, a shared authentication component of a cloud computing system used to provide multiple different business resources, such as email applications and case management systems, to multiple different users 601. As will be described hereafter, in general, apparatus 600 may include authentication functionality. In particular, apparatus 600 may validate authentication requests sent to apparatus 600 when a user 601 requests authentication. Such validation of an authentication request may be separate from the processing of an authentication request.

[00065] The terms “authenticate” and “authentication” as used herein refer to establishing an identity for a user 601 based on, for example, credentials provided by the user or computing device (e.g., username, password, etc.). Such credentials may be provided as part of an authentication request within a communication, and authentication includes the processing of such credentials within an authentication request to establish an identity. A user 601 whose identity has been established by an authentication process may sometimes be referred to herein as "authenticated" or an "authenticated user."

[00066] The terms "validate" and "validation" as used herein with reference to authentication requests refers to vetting authentication request communications sent to apparatus 600 to determine whether the authentication request contained therein is accepted (resulting in a "validated request") or rejected (resulting in a request that is "not validated"), as opposed to processing the actual authentication requests contained within such communications in order to establish an identity. Such communications may include, for example, user identifier data that is out of band with respect to the authentication request, and validation includes, for example, the processing of such user identifier data.

[00067] While the examples disclosed herein are primarily described in the context of establishing an identity for a user, it is contemplated the establishment of such an identity may include establishing the identity of a computing device that may be operated by and/or associated with a user, or an autonomous computing system or device, and as such, the term "user" in this context is not limited to persons. It will also be understood that the examples described herein "establish an identity" for a user based on certain data and credentials provided by the user, as opposed to a positive physical identification of the user actually providing such data and credentials, and it is assumed that users possessing and presenting such data and credentials have obtained the appropriate permissions to do so. As such, while the examples described herein may provide increased security, the term "establish an identity" should not be construed as guaranteeing a positive physical identification of a user presenting such data and credentials for purposes of authentication.

[00068] Apparatus 600 may include processing electronics 602. Processing electronics 602 may be similar to processing electronics 102 as shown in and described with reference to Figure 1. For example, as shown in Figure 6, processing electronics 602 may include a processor 604 configured to execute logic in the form of instruction modules contained in a memory 606, similar to processor 104 and memory 106 as shown in and described with reference to Figure 1. As with processor 104 and memory 106, processor 604 and memory 606 may be single devices, or may comprise multiple processors and memories having distributed functionality.

[00069] Memory 606 may include records 608 stored therein. Records 608 may be similar to records 108 as shown in and described with reference to Figure 1. For example, records 608 may include user data items for multiple different users and may be defined by and stored using correlated tree structures 610 similar to tree structures 110 as shown in and described with reference to Figure 1. In particular, tree structures 610 may have schema definitions such that stored records 608 are unrestricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns.

[00070] Memory 606 may also include privacy labels 612 stored therein. Privacy labels 612 may be similar to privacy labels 112 as shown in and described with reference to Figure 1. For example, privacy labels 612 may distinguish among multiple different users 601. Each privacy label 612 may include a unique identification mapping to a user 601. In some examples, privacy label 612 may include a number of tokens (e.g., unique randomly generated cryptographically entropic values having a predetermined number of bytes, etc.) that are uniquely provisioned to user 601. Privacy labels 612 may be correlated with each of records 608 using privacy label tree structures included in tree structures 610. The privacy label tree structures defining privacy labels 612 may have schema definitions similar to other tree structures 610. That is, privacy labels 612 may be structured and stored without being restricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns. Privacy labels 612 may be compared with user identifier data provided by users 601 in order to facilitate validation of authentication requests. In some examples, privacy labels 612 may include local copies of privacy labels 612 that may be stored in a remote location along with records 608. For example, records 608 and privacy labels 612 may be stored and maintained in a remote and/or separate multiple user data storage and separation system, with a local copy of privacy labels stored in memory 606 of apparatus 600.

[00071] Memory 606 may also include receiving module 614, label identification module 616, and validation module 620. Modules 614, 616, and 620 may cooperate to cause processing electronics 602 to carry out the process 700 set forth by the flow diagram of Figure 7. As indicated by a step 702, receiving module 614 may receive a communication 621 including user identifier data 622 and authentication request 624. User identifier data 622 may include data correlated with privacy labels 612, and each privacy label 612 may include a unique identification mapping to a user. For example, a privacy label 612 may include a number of tokens that are uniquely provisioned to user 601. User 601 may send a communication 621 including user identifier data 622 that includes a set of N tokens that have been provisioned to user 601. User identifier data 622 may be included in communication 621 along with authentication request 624. User identifier data 622 may be used by apparatus 600 to validate authentication request 624 (e.g., prior to or during processing of authentication request 624).

[00072] In some examples, user 601 may be prompted to provide input (e.g., using a client computing device in communication with apparatus 600) that determines the content of user identifier data 622. For example, user 601 may have been provisioned tokens for multiple different users (e.g., tokens that allow user 601 to access business resources and manage data for multiple different entities). User 601 may be prompted and/or required to select whether to request authentication for one of the entities (e.g., one or several entities have requested that separate authentication be performed prior to accessing its business resources and/or data) or several of the entities (e.g., user 601 may perform batch data requests including data for multiple entities using a single authentication). User identifier data 622 may be populated with the appropriate tokens depending on the input received from user 601. In some examples, user 601 may select a default set of user identifier data 622, and may be prompted to add tokens in order to authenticate for additional business resources and/or users for which user 601 has been provisioned tokens. In some examples, user 601 may be permitted to authenticate for multiple business resources and users at once (e.g., simultaneously or via subsequent additional authentication requests). In some examples, user 601 may only be permitted to authenticate for only one resource or entity at a time.

[00073] In some examples, user identifier data 622 may be out of band data with respect to authentication request 624. That is, user identification data 622 may be received by receiving module 614 in the same communication as authentication request 624, but may be kept separate from authentication request 624 by a conceptually independent data channel provided as an inherent characteristic of the communication channel and transmission protocol, as opposed to requiring a separate communication channel and endpoints to be established at apparatus 600. As such, user identifier data 622 may be used to validate authentication request 624 independent of processing authentication request 624. Authentication request 624 may be, for example, a query that defines the scope of an authentication request. For example, authentication request 624 may include authentication credentials for a user 601, such as a username and/or password, along with a request to authenticate the identity of user 601.

[00074] As will be appreciated, providing user identification data 622 in combination with authentication request 624 in communication 621 may facilitate “single sign on” authentication for multiple business resources. That is, user identification data 622 may be used for validation of authentication request communications, and to distinguish among multiple entities or other users and the resources and records to which they are associated. Credentials and other information included in authentication request 624 may be independently used solely to establish the identity of a particular user or computing device associated with user identification data 622 during authentication, as opposed to also distinguishing among multiple entities and resources to which user 601 may be authenticated. User 601 may utilize the same credentials (e.g., username and password) for multiple business resources and/or users for which user 601 has been provisioned unique user identification data 622 while still maintaining data separation among such business resources and/or users. As will also be appreciated, using user identification data 622 in combination with authentication request 624 in communication 621 may provide for more robust security. That is, even if an observer of network traffic were to obtain user 601's confidential credentials, the observer will not be able to utilize these credentials to authenticate its identity with apparatus 601 without also obtaining user 601's user identification data 622.

[00075] At a step 704, label identification module 616 may identify a set of privacy labels 612 based on user identifier data 622. In some examples, label identification module 616 may identify the set of the privacy labels 612 by comparing tokens in the user identifier data 622 with tokens in privacy labels 612 stored in memory 606. In some examples, label identification module 616 may communicate with a privacy label mapping service to obtain the actual tokens for the ones provided by user 601 (e.g., user 601 is not provided with the actual tokens used by apparatus 600, but rather reference tokens) in order to identify the set of privacy labels. If, for example, tokens in the user identifier data 622 provided by user 601 do not match tokens in any of privacy labels 612, the set of privacy labels identified by label identification module 616 will be an empty set. If tokens in the user identifier data 622 provided by user 601 match tokens privacy labels 612, the set of privacy labels identified by label identification module 616 will not be an empty set.

[00076] At a step 706, validation module 620 may determine whether to validate authentication request 624 based on the set of privacy labels 612. In some examples, if validation module 620 determines that the set of privacy labels 612 identified by label identification module 616 is an empty set, then validation module 620 may determine that authentication request 624 is not validated (i.e., rejected). Similarly, if validation module 620 determines that the set of privacy labels 612 identified by label identification module 616 is not an empty set, then validation module 620 may determine that authentication request 624 is validated (i.e., accepted). In some examples, additional security metrics may be evaluated prior to validation of the authentication request by validation module 620.

[00077] At a step 708, validation module 620 may provide a response to authentication request 624. For example, if validation module 620 determines that authentication request 624 is not validated, then user 601 may be sent a response to the authentication request indicating that the request is not validated. If validation module 620 determines that authentication request 624 is validated, then user 601 may be sent a response to the authentication request indicating that the request is validated, and that apparatus 600 will proceed with processing the authentication request. For example, validation module 620 may provide a nonce to user 601 in the response to the authentication request. The nonce may be encrypted by user 601 and sent back to apparatus 600 for further processing as part of an authentication process to establish the identity of user 601.

[00078] Figure 8 is a data flow diagram of another example authentication apparatus 800 for use with a multiple user data storage and separation apparatus, such as apparatus 100 shown in Figure 1. As will be described hereafter, in general, apparatus 800 may include authentication functionality. In particular, apparatus 800 may validate authentication requests based on user identifier data included in communications that embody such authentication requests. Apparatus 800 is similar to apparatus 600 shown in and described with reference to Figure 6, except that apparatus 800 may additionally include security module 830, mapping module 832, authentication module 834, session module 836, and authorization module 838. Those remaining components of apparatus 800 that correspond to apparatus 600 are numbered similarly. In some examples, modules 614, 616, 620, and 830 may be integrated into a network facing module, while modules 832, 834, 836, and 838 may be accessible only by the network facing module to prevent privacy labels 612 and unencrypted user credentials from being misappropriated during authentication. In some examples, session module 836 and authorization module 838 may be provided as a separate systems accessible by apparatus 800.

[00079] Modules 614, 616, 620, 830, 832, 834, 836, and 838 may cooperate to cause processing electronics 602 to carry out the process 900 set forth by the flow diagram of Figure 9. As indicated by a step 902, receiving module 614 may receive a communication 621 including user identifier data 622 and authentication request 624. User identifier data 622 may be used by apparatus 600 to validate authentication request 624 (e.g., prior to or during processing of authentication request 624). User identifier data 622 may include data correlated with a set of privacy labels 612, and each privacy label 612 may include a unique identification mapping to a user. For example, a privacy label 612 may include a number of tokens that are uniquely provisioned to user 601. The tokens may be, for example, a random number generated using, for example, a random number generator compliant with Federal Information Processing Standard (FIPS) Publication 140-2/3. In some examples, the tokens may include a minimum of 256 random bits. User 601 may send a communication 621 including user identifier data 622 that includes a set of N tokens that have been provisioned to user 601, and which may be correlated with the privacy label 612. The user identifier data 622 including the set of N tokens may be included in communication 621 along with authentication request 624.

[00080] In some examples, user 601 may not be provisioned with the actual tokens used by apparatus 600, but rather reference tokens that may be mapped to the actual tokens used in privacy labels 612. Reference tokens may be used between user 601 and any network facing modules to, for example, prevent an observer of network traffic from obtaining the actual privacy labels used for separation and storage of data. In such examples, user identification data 622 may include the reference tokens, which may be mapped to the actual tokens using mapping module 832 at a step 904. For example, user 601 may be provisioned a reference token having the value “1111 … 11”, which may be correlated with a set privacy labels 612 having the actual values “8A8A … 8A”, “8B8B … 8B”, and “8C8C … 8C”. In some examples, intermediate privacy label mappings may be used within apparatus 800. Intermediate privacy label mappings may be used to limit access privileges of a user 601 before that user has been appropriately authenticated by apparatus 800. For example, for purposes of validation and/or authentication, the reference token value “1111 … 11” provisioned to user 601 may be mapped to an intermediate reference privacy label including the value “2222 … 22”. The intermediate privacy label value “2222 … 22” may be, for example, a generic privacy label that allows user 601 to communicate with authentication module 834 via network facing modules during authentication, but prevents user 601 from reading, writing, updating, deleting, or purging records 608.

[00081] In some examples, user identifier data 622 may be out of band data with respect to authentication request 624. That is, user identification data 622 may be received by receiving module 614 in the same communication as authentication request 624, but may be kept separate from authentication request 624 by a conceptually independent data channel provided as an inherent characteristic of the communication channel and transmission protocol, as opposed to requiring a separate communication channel and endpoints to be established at apparatus 800. As such, user identifier data 622 may be used to validate authentication request 624 independent of processing authentication request 624. Authentication request 624 may be, for example, a query that defines the scope of an authentication request. For example, authentication request 624 may include data or credentials for a user 601, such as a username and/or password, along with a request to authenticate the identity of user 601.

[00082] At a step 906, label identification module 616 may identify a set of privacy labels 612 based on user identifier data 622. In some examples, label identification module 616 may identify the set of the privacy labels 612 by comparing tokens in the user identifier data 622 with tokens in privacy labels 612 stored in memory 606. In some examples, at step 904, label identification module 616 may communicate with a privacy label mapping service provided by mapping module 832 to obtain the actual tokens for the ones provided by user 601. If, for example, tokens in the user identifier data 622 provided by user 601 do not match tokens in any of privacy labels 612, the set of privacy labels identified by label identification module 616 will be an empty set. If tokens in the user identifier data 622 provided by user 601 match tokens privacy labels 612, the set of privacy labels identified by label identification module 616 will not be an empty set.

[00083] For example, user 601 may be provisioned a reference token having the value “1111 … 11”, which may be included in user identifier data 622. Label identification module 616 may communicate with mapping module 832 to identify the set of privacy labels 612. In particular, label identification module 616 and mapping module 832 may receive the reference token value “1111 … 11” and identify the set of privacy labels 612 having the actual values “8A8A … 8A”, “8B8B … 8B”, and “8C8C … 8C”. As such, the set of privacy labels 612 is not an empty set. For purposes of validation and authentication, however, the reference token value “1111 … 11” provisioned to user 601 may be mapped to the reference privacy label value “2222 … 22”. In contrast, a user 601 may provide a reference token having the value “0101 … 01”, which may be a false or de-provisioned value that is included in user identifier data 622. Label identification module 616 and mapping module 832 may receive the reference token value “0101 … 01”, and determine that no privacy labels 612 include matching tokens for the value “0101 … 01”. As such, the set of privacy labels 612 is an empty set.

[00084] At a step 908, security metrics module 830 may execute a number of perimeter security metrics checks that may be used to validate the authentication request. The security checks may utilize data indicative of the context of communication 621. For example, security metrics module 830 may use geographic location data for a user 601 associated with the authentication request in order to execute a security check. Security metrics module 830 may derive geolocation data from an IP address encapsulated in communication 621 and apply user policies to determine if the authentication request has originated from an improper or excluded location. In some examples, security metrics module 830 may use temporal data encapsulated in communication 621 in order to perform a security check. Security metrics module 830 may determine what time communication 621 was originated and apply user policies to determine if the authentication request has originated during an improper or excluded time period.

[00085] In some examples, security metrics module 830 may perform a number of enumeration checks that may be used to validate the authentication request. For example, security metrics module 830 may determine the frequency of requests for a user 601 associated with the authentication request and apply user policies to determine if user 601 is abusing the system by exceeding certain thresholds. In some examples, security metrics module 830 may determine a number of, or the frequency of, prior non-validated authentication requests for a user 601 associated with the authentication request and compare it with certain thresholds to determine whether the authentication request is genuine or whether it may indicate an attempted breach of security. In some examples, security metrics module 830 may perform a comparison of a number of prior non-validated authentication requests with a number of prior validated authentication requests for a user associated with the validation request and compare it with certain thresholds to determine whether user 601 is experiencing technical difficulties or whether it may indicate an attempted breach of security, an attempt to disrupt operation of apparatus 600 and any associated system components, etc.

[00086] In some examples, security metrics module 830 may adjust a threshold for the number of, or frequency of, non-validated authentication requests that may be accumulated by a user 601 based on a number of non-validated authentication requests received from users in multiple different locations. For example, if a certain number of non-validated authentication requests are received from multiple geographic locations within a particular period of time, security metrics module 830 may temporarily modify acceptable thresholds for the number of, or frequency of, non-validated authentication requests. An increase in non-validated authentication requests that are received from multiple geographic locations within a particular period of time may indicate a broad-based attempted security breach, and security metrics module 830 may accordingly lower certain thresholds for the number of, or frequency of, non-validated authentication requests. Similarly, a decrease in non-validated authentication requests that are received from multiple geographic locations within a particular period of time may cause security metrics module 830 to raise certain thresholds for the number of, or frequency of, non-validated authentication requests to normal acceptable levels.

[00087] At a step 910, validation module 620 may determine whether to validate authentication request 624 based on the set of privacy labels 612. In some examples, if validation module 620 determines that the set of privacy labels 612 identified by label identification module 616 is an empty set, then validation module 620 may determine that authentication request 624 is not validated (i.e., rejected). Similarly, if validation module 620 determines that the set of privacy labels 612 identified by label identification module 616 is not an empty set, then validation module 620 may determine that authentication request 624 is validated (i.e., accepted). In some examples, security metrics checks performed by security metrics module 830 may be used by validation module 620 to determine whether to validate the authentication request. For example, even if validation module 620 determines that the set of privacy labels 612 identified by label identification module 616 is not an empty set, security metrics module 830 may indicate that the authentication request has failed a security metrics check. Validation module 620 may accordingly determine that authentication request 624 is not validated. In some examples, each instance of a non-validated authentication request for a user may be recorded by security metrics module 830 for use in evaluating and determining the security metrics described above.

[00088] If validation module 620 has determined that authentication request 624 is not validated, then at a step 912, user 601 may be sent a response to authentication request 624 indicating that the request is not validated. If validation module 620 determines that authentication request 624 is validated, then at a step 914, user 601 may be sent a response to authentication request 624 indicating that the request is validated, and that apparatus 600 will proceed with processing authentication request 624. For example, at step 914, validation module 620 may provide a nonce to user 601 in the response to the authentication request. The nonce may be, for example, a random number generated by authentication module 834 using, for example, a random number generator compliant with Federal Information Processing Standard (FIPS) Publication 140-2/3. In some examples, the nonce may include a minimum of 256 random bits.

[00089] At a step 916, receiving module 614 may receive a communication 840 including user identifier data 842 and authentication request 844. Similar to communication 622, communication 840 may include user identifier data 842 that includes a set of N tokens along with an authentication request to apparatus 600. Authentication request 844 may include an encrypted version of the nonce. Steps 918-924 may essentially repeat steps 904-910. For example, at a step 918, mapping module 918 may map reference tokens in user identifier data 842 to actual tokens, and at a step 920, label identification module 616 may identify a set of privacy labels 612 based on user identifier data 842. At a step 922, security metrics module 830 may execute a number of perimeter security metrics checks that may be used to validate authentication request 844. At a step 924, validation module 620 may determine whether to validate authentication request 844 based on the set of privacy labels 612 and, in some examples, security metrics checks performed by security metrics module 830 may also be used by validation module 620 to determine whether to validate authentication request 844.

[00090] If validation module 620 has determined not to validate authentication request 844, then at a step 926, user 601 may be sent a response to authentication request 844 indicating that the request is not validated. If validation module 620 determines to validate authentication request 844, then at a step 928, validation module 620 may provide the encrypted nonce included in authentication request 844 to authentication module 834. At a step 930, authentication module 834 may execute an authentication process using the encrypted version of the nonce to determine whether to confirm an identity for a user 601 associated with authentication request 844 based on credentials included in authentication request 844. For example, authentication module 834 may decrypt the encrypted version of the nonce using a hash. The hash may be, for example, a hash generated by a mutually defined hash function and applied to keying material (e.g., a password) for a user 601 associated with authentication request 844. The hash may be used by user 601 to encrypt the nonce, and by authentication module 834 to decrypt the nonce.

[00091] If authentication module 834 determines that an identity for user 601 is not confirmed, then at a step 932, validation module 620 may provide a response 846 to user 601 indicating that authentication has failed. If authentication module 834 determines that an identity for user 601 has been established based on credentials included in authentication request 844, then at a step 934, validation module 620 may provide a response 846 to user 601 indicating that user 601 has been authenticated. In some examples, response 846 may include a session identifier generated by session module 836. The session identifier may be assigned to user 601. The session identifier may indicate that user 601 has been authenticated. For example, the session identifier may be a session token correlated with tokens provisioned to user 601 (e.g., tokens provided in user identifier data 622 and/or 840) and used by apparatus 800 to verify that an identity for user 601 has been established. The session token may be, for example, a random number generated by session module 836 using, for example, a random number generator compliant with Federal Information Processing Standard (FIPS) Publication 140-2/3. In some examples, the session token may include a minimum of 256 random bits. In some examples, the session token may be correlated with a reference privacy label 612 that is used for purposes of validation and/or authentication. For example, the session token may be correlated with a reference privacy label 612 including the token value "2222 … 22" that is also mapped to a token value "1111 … 11" provisioned to user 601.

[00092] The session identifier may also function as a confirmation of the authentication of user 601 that may be accessible by multiple different business resources 850 in order to facilitate cross-domain authorization. Business resources 850 may be, for example, multiple different resources that utilize records 608 that are correlated with user 601 using a set of privacy labels 612. Upon receiving a communication including a data request from user 601, a business resource 850 may check to see if a valid session identifier is correlated with user identifier data provided by user 601 (e.g., tokens provisioned to user 601) along with the request for data.

[00093] In some examples, validation module 620 may communicate with authorization module 838 to indicate that that user 601 has been authenticated and has been assigned a valid session identifier. Authorization module 838 may function as an intermediary between user 601 and resources 850 such that user 601 may access only those resources 850 utilizing records 608 for which user 601 has been provisioned (e.g., records 608 having correlated privacy labels 612 that are mapped to tokens provisioned to user 601). Authorization module 838 may also provide a central location where resources 850 may access a confirmation that user 601 has been appropriately provisioned and that user 601 has been authenticated.

[00094] For example, in response to receiving an indication that user 601 has been authenticated and has been assigned a valid session identifier, authorization module 838 may provide an authorization identifier to validation module 620. The authorization identifier may be in authorization token, which may be, for example, a random number generated by authorization module 838 using, for example, a random number generator compliant with Federal Information Processing Standard (FIPS) Publication 140-2/3. In some examples, the authorization token may include a minimum of 256 random bits. In some examples, the authorization token may be correlated with a reference privacy label 612 that is used for purposes of validation and/or authentication. For example, the authorization token may be correlated with a reference privacy label 612 including the token value “2222 … 22” that is also mapped to a token value “1111 … 11” provisioned to user 601. The reference privacy label 612 may also be correlated with a session identifier. The authorization identifier may also be provided to user 601 in response 846 along with the session identifier and a resource locator. The term “resource locator” as used herein refers to data, such as an Internet Protocol (IP) address (e.g., IPv4, IPv6, etc.) or other identifier that includes a name, location, and or route, indicating how user 601 may access a resource, such as authorization module 838. User 601 may then access authorization module 838 to request access to business resources 850 using the session identifier and the authorization identifier, and authorization module 838 may use the session identifier and the authorization identifier to confirm for business resources 850 that user 601 has been appropriately provisioned and that user 601 has been authenticated.

[00095] Figure 10 is a schematic illustration of an example authorization apparatus 1000 for use with a multiple user data storage and separation apparatus, such as apparatus 100 shown in Figure 1. Apparatus 1000 may be, for example, a shared authorization component of a cloud computing system used to provide multiple different business resources, such as email applications and case management systems, to multiple different users 1001. As will be described hereafter, in general, apparatus 1000 may include authorization functionality. In particular, apparatus 1000 may process authorization requests and generate a set of permissions that define a scope of access for a user to a resource. The terms “authorize” and ‘authorization” as used herein refer to the process of associating permissions that define a scope of access to a particular resource with a user 1001. As with authentication, while the examples disclosed herein are primarily described in the context of associating permissions with a user, it is contemplated the association of such permissions may include associating permissions with a computing device that may be operated by and/or associated with a user, or an autonomous computing system or device, and as such, the term “user” in this context is not limited to persons. The terms “validate” and “validation” as used herein with reference to authorization requests refers to vetting authorization request communications sent to apparatus 1000 to determine whether the authorization request contained therein is accepted (resulting in a “validated request”) or rejected (resulting in a request that is “not validated”), as opposed to processing the actual authorization requests contained within such communications in order to associate permissions with a user 1001.

[00096] Apparatus 1000 may include processing electronics 1002. Processing electronics 1002 may be similar to processing electronics 102 as shown in and described with reference to Figure 1. For example, as shown in Figure 10, processing electronics 1002 may include a processor 1004 configured to execute logic in the form of instruction modules contained in a memory 1006, similar to processor 104 and memory 106 as shown in and described with reference to Figure 1. As with processor 104 and memory 106, processor 1004 and memory 1006 may be single devices, or may comprise multiple processors and memories having distributed functionality.

[00097] Memory 1006 may include records 1008 stored therein. Records 1008 may be similar to records 108 as shown in and described with reference to Figure 1. For example, records 1008 may include user data items for multiple different users and may be defined by and stored using correlated tree structures 1010 similar to tree structures 110 as shown in and described with reference to Figure 1. In particular, tree structures 1010 may have schema definitions such that stored records 1008 are unrestricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns.

[00098] Memory 1006 may also include privacy labels 1012 stored therein. Privacy labels 1012 may be similar to privacy labels 112 as shown in and described with reference to Figure 1. For example, privacy labels 1012 may distinguish among multiple different users 1001. Each privacy label 1012 may include a unique identification mapping to a user 1001. In some examples, privacy label 1012 may include a number of tokens (e.g., unique randomly generated cryptographically entropic values having a predetermined number of bytes, etc.) that are uniquely provisioned to user 1001. Privacy labels 1012 may be correlated with each of records 1008 using privacy label tree structures included in tree structures 1010. The privacy label tree structures defining privacy labels 1012 may have schema definitions similar to other tree structures 1010. That is, privacy labels 1012 may be structured and stored without being restricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns. Privacy labels 1012 may be compared with user identifier data provided by users 1001 in order to facilitate validation and processing of authentication requests. In some examples, privacy labels 1012 may include local copies of privacy labels 1012 that may be stored in a remote location along with records 1008. For example, records 1008 and privacy labels 1012 may be stored and maintained in a remote and/or separate multiple user data storage and separation system, with a local copy of privacy labels stored in memory 1006 of apparatus 1000.

[00099] Memory 1006 may also include receiving module 1014, label identification module 1016, and validation module 1020. Modules 1014, 1016, and

1020 may cooperate to cause processing electronics 1002 to carry out the process 1100 set forth by the flow diagram of Figure 11. As indicated by a step 1102, receiving module 1014 may receive a communication 1021 including user identifier data 1022 and authentication request 1024. User identifier data 1022 may include data correlated with privacy labels 1012, and each privacy label 1012 may include a unique identification mapping to a user. For example, a privacy label 1012 may include a number of tokens that are uniquely provisioned to user 1001. User 1001 may send a communication 1021 including user identifier data 1022 that includes a set of N tokens that have been provisioned to user 1001. User identifier data 1022 may be included in communication 1021 along with authorization request 1024. In some examples, user identifier data 1022 may be used by apparatus 1000 to validate authorization request 1024 (e.g., prior to or during processing of authorization request 1024). In some examples, user identifier data may also include a session identifier and/or an authentication identifier (e.g., provided to user 1001 during authentication) that may be used by apparatus 1000 to validate authentication request 1024.

[000100] At a step 1104, label identification module 1116 may identify a set of privacy labels 1012 based on user identifier data 1022. In some examples, label identification module 1016 may identify the set of the privacy labels 1012 by comparing tokens in the user identifier data 1022 with tokens in privacy labels 1012 stored in memory 1006. In some examples, label identification module 1016 may communicate with a privacy label mapping service to obtain the actual tokens for the ones provided by user 1001 (e.g., user 1001 is not provided with the actual tokens used by apparatus 1000, but rather reference tokens) in order to identify the set of privacy labels.

[000101] At a step 1106, authorization module 1020 may generate a set of corresponding permissions 1023 for each privacy label in the set. Each set of corresponding permissions 1023 may define a scope of access for user 1001 to a resource 1050 corresponding to each respective privacy label 1012 in the set. For example, a newly deployed resource 1050 may define a number of possible roles (e.g., manager, administrator, auditor, employee, etc.) that may be provisioned to new users by authorization module 1020. Authorization module 1020 may variously provision the roles to a number of users by correlating a role with privacy label 1012 that has been provisioned to the user. A list of roles correlated with privacy labels 1012 may be stored in memory 1006. Each role may include a specific set of permissions 1023 and the identity of a particular resource 1050 that may be accessed by user 1001.

[000102] For each privacy label 1012 in the set of privacy labels 1012, authorization module 1020 may identify a corresponding role and generate the set of permissions 1023. Depending on the desired scope of access for user 1001, the set of permissions 1023 may include, for example, a read permission, a write permission, an update permission, a delete permission, and/or a purge permission. A read permission may be, for example, a permission that allows or denies user 1001 read-only access to records 1008 associated with the corresponding privacy label 1012. A write permission may be, for example, a permission that allows or denies user 1001 the ability to add records 608 associated with the corresponding privacy label 1012. An update permission may be, for example, a permission that allows or denies user 1001 the ability to modify records 608 associated with the corresponding privacy label 1012. A delete permission may be, for example, a permission that allows or denies user 1001 the ability to delete, but not remove, records 608 associated with the corresponding privacy label 1012. A purge permission may be, for example, a permission that allows or denies user 1001 the ability to remove deleted records 608 associated with the corresponding privacy label 1012. The set of permissions 1023 may also define the user as a person, a system, or a proxy for a person or system.

[000103] At a step 1108, authorization module 1020 may provide a privacy label 1012 from the set of privacy labels 1012 and its corresponding set of permissions 1023 to a resource 1050 corresponding to the privacy label in a communication 1027. In some examples, authorization module 1020 may regenerate a particular set of permissions 1023 and provide it to the resource 1050. For example, authorization module 1020 may de-provision a role or reassign a role to a user 1001. Authorization module 1020 may regenerate the set of permissions 1023 and provide it to resource

1050 in order to reflect the change. For example, authorization module 1020 may perform a regeneration periodically (e.g., using a timer), upon a triggering event (e.g., a data request), etc.

[000104] At a step 1110, authorization module 1020 may provide a response to authorization request 1024. Response 1024 may include, for example, a resource locator for each resource 1050 corresponding to a privacy label 1012 in the set of privacy labels 1012 for which a set of permissions 1023 was generated. User 1001 may then send data requests to resource 1050 using the resource locator. In some examples, such data requests may be sent to and processed by resource 1050 in a manner similar to that described above with reference to Figure 4 above. Resource 1050 may utilize the set of permissions received in step 1108 to determine whether user 1001 has a permission required for the data request (e.g., read access to a particular data record included in the data request, etc.).

[000105] In this way, apparatus 1000 may define access permissions for authenticated users that are custom tailored for each resource such that they may be neither under-inclusive nor over-inclusive. By correlating sets of permissions with privacy labels and appropriately provisioning permissions and privacy labels to each user, improved granularity in assigning access permissions during authorization may be achieved, and a single authentication system may be used across multiple different business resources for multiple different entities. Permissions for a particular user with respect to a particular resource may be adjusted “on-the-fly” in order to reflect recent changes.

[000106] Figure 12 is a schematic illustration of another example authorization apparatus 1200 for use with a multiple user data storage and separation apparatus, such as apparatus 100 shown in Figure 1. As will be described hereafter, in general, apparatus 1200 may include authorization functionality. In particular, apparatus 1200 may process authorization requests and generate a set of permissions that define a scope of access for a user to a resource. Apparatus 1200 is similar to apparatus 1000 shown in and described with reference to Figure 10, except that apparatus 1200 may additionally include mapping module 1232 and session module 1236. Session module

1232 and mapping module 1236 may be similar to session module 832 and mapping module 836 as shown in and described with respect to Figure 8. In some examples, authorization module 1020, session module 1232 and mapping module 1236 may provide functionality for both authorization apparatus 1200 and an authentication apparatus, such as apparatus 800 shown in Figure 8 (e.g., in place of authorization module 838, session module 832, and mapping module 836. Those remaining components of apparatus 1200 that correspond to apparatus 1000 are numbered similarly. In some examples, modules 1014, 1016, and 1020 may be integrated into a network facing module, while modules 1232 and 1236 may be accessible only by the network facing module.

[000107] Modules 1014, 1016, 1020, 1232, and 1236 may cooperate to cause processing electronics 1002 to carry out the process 1300 set forth by the flow diagram of Figure 13. As indicated by a step 1302, receiving module 1014 may receive a communication 1021 including user identifier data 1022 and authorization request 1024. For example, user 1001 may have successfully completed an authentication process and may have further received information on how user 1001 may access authorization apparatus 1200 (e.g., an IP address). User identifier data 1022 may include data correlated with a set of privacy labels 1012, and each privacy label 1012 may include a unique identification mapping to a user. For example, a privacy label 1012 may include a number of tokens that are uniquely provisioned to user 1001. The tokens may be, for example, a random number generated using, for example, a random number generator compliant with Federal Information Processing Standard (FIPS) Publication 140-2/3. In some examples, the tokens may include a minimum of 256 random bits. User 1001 may send a communication 1021 including user identifier data 622 that includes a set of N tokens that have been provisioned to user 1001, and which may be correlated with the privacy label 1012. The user identifier data 1022 including the set of N tokens may be included in communication 1021 along with authorization request 1024.

[000108] In some examples, user 1001 may not be provisioned with the actual tokens used by apparatus 1200, but rather reference tokens that may be mapped to the actual tokens used in privacy labels 1012. Reference tokens may be used between user 1001 and any network facing modules to, for example, prevent an observer of network traffic from obtaining the actual privacy labels used for separation and storage of data. In such examples, user identification data 1022 may include the reference tokens, which may be mapped to the actual tokens using mapping module 1232 at a step 1304. For example, user 1001 may be provisioned a reference token having the value “1111 … 11”, which may be correlated with a set privacy labels 612 having the actual values “8A8A … 8A”, “8B8B … 8B”, and “8C8C … 8C”. In some examples, intermediate privacy label mappings may be used within apparatus 1200. Intermediate privacy label mappings may be used to limit access privileges of a user 1201 during authorization by apparatus 1300. For example, for purposes of authentication and authorization, the reference token value “1111 … 11” provisioned to user 1001 may be mapped to an intermediate reference privacy label including the value “2222 … 22”. The intermediate privacy label value “2222 … 22” may be, for example, a generic privacy label that allows user 1001 to communicate with apparatus 1200 via network facing modules during authorization, but prevents user 1001 from reading, writing, updating, deleting, or purging records 608.

[000109] In some examples, user identifier data 1022 may be out of band data with respect to authorization request 1024. That is, user identification data 1022 may be received by receiving module 1014 in the same communication as authentication request 1024, but may be kept separate from authentication request 1024 by a conceptually independent data channel provided as an inherent characteristic of the communication channel and transmission protocol, as opposed to requiring a separate communication channel and endpoints to be established at apparatus 1200. As such, user identifier data 622 may be used to validate authorization request 1024 independent of processing authorization request 1024.

[000110] In some examples, user identifier data may also include a session identifier and/or an authentication identifier that may be used by apparatus 1000 to validate authentication request 1024. For example, the session identifier and/or the authentication identifier may be generated and provided during authentication as described above with reference to Figure 8, and with particular reference to session module 836 and authorization module 838. Authorization request 1024 may be, for example, a query that defines the scope of an authentication request.

[000111] At a step 1306, label identification module 1016 may identify a set of privacy labels 1012 based on user identifier data 1022. In some examples, label identification module 1016 may identify the set of the privacy labels 1012 by comparing tokens in the user identifier data 1022 with tokens in privacy labels 1012 stored in memory 1006. In some examples, at step 1304, label identification module 1016 may communicate with a privacy label mapping service provided by mapping module 1232 to obtain the actual tokens for the ones provided by user 1001 prior to comparing with the tokens in privacy labels 1012. If the actual tokens do not match tokens in any of privacy labels 1012, the set of privacy labels identified by label identification module 1016 will be an empty set. If the actual tokens match tokens privacy labels 1012, the set of privacy labels identified by label identification module 1016 will not be an empty set.

[000112] For example, user 1001 may be provisioned a reference token having the value "1111 … 11", which may be included in user identifier data 1022. Label identification module 1016 may communicate with mapping module 1232 to identify the set of privacy labels 1012. In particular, label identification module 1016 and mapping module 1232 may receive the reference token value "1111 … 11" and identify the set of privacy labels 1012 having the actual values "8A8A … 8A", "8B8B … 8B", and "8C8C … 8C". As such, the set of privacy labels 1012 is not an empty set. For purposes of authorization, however, the reference token value "1111 … 11" provisioned to user 1001 may be mapped to the reference privacy label value "2222 … 22". In contrast, a user 1001 may provide a reference token having the value "0101 … 01", which may be a false or de-provisioned value that is included in user identifier data 1022. Label identification module 1016 and mapping module 1232 may receive the reference token value "0101 … 01", and determine that no privacy labels 1012 include matching tokens for the value "0101 … 01". As such, the set of privacy labels 1012 is an empty set.

[000113] At a step 1308, authorization module 1020 and/or session module 1236 may verify the authorization identifier and/or the session identifier provided in user identifier data 1022. Authorization module 1020 and/or session module 1236 may verify the authorization identifier and/or the session identifier by comparing them with stored values correlated with a reference privacy label 612 including the token value "2222 … 22" that is also mapped to a token value "1111 … 11" provisioned to user 601. The values of the authorization identifier and/or the session identifier provided in user identifier data 1022 may be verified if they match the stored values.

[000114] At a step 1310, authorization module 1020 may determine whether to validate authorization request 1024 based on the set of privacy labels 1012. In some examples, if authorization module 1020 determines that the set of privacy labels 1012 identified by label identification module 1016 is an empty set, then authorization module 1020 may determine that authorization request 1024 is not validated (i.e., rejected). Similarly, if authorization module 1020 determines that the set of privacy labels 1012 identified by label identification module 1016 is not an empty set, then authorization module 1020 may determine that authorization request 1024 is validated (i.e., accepted). In some examples, the result of a verification of an authorization identifier and/or a session identifier at step 1308 may be used by validation module 620 to determine whether to validate the authorization request. For example, even if authorization module 1020 determines that the set of privacy labels 1012 identified by label identification module 1016 is not an empty set, authorization module 1020 and/or session module 1236 may have determined at step 1308 that either an authorization identifier and/or a session identifier has not been verified. Authorization module 1020 may accordingly determine that authorization request 1024 is not validated. If authorization module 1020 has determined that authorization request 1024 is not validated, then at a step 1312, user 1001 may be sent a response to authorization request 1024 indicating that access is denied. If authorization module 1020 determines that authorization request 1024 is validated, then apparatus 1200 may proceed with processing authorization request 1024.

[000115] At a step 1314, authorization module 1020 and mapping module 1232 may generate a map of each privacy label 1012 in the set identified by label identification module 1016 to a corresponding reference privacy label. In some examples, each corresponding reference privacy label generated in the map may be an updated reference privacy label to be used in place of an intermediate privacy label for purposes of data requests made by user 1001. For example, user 1001 may be provisioned a reference token having the value "1111 … 11", which may be included in user identifier data 1022. Label identification module 1016 may communicate with mapping module 1232 at steps 1304 and 1306 to identify the set of privacy labels 1012. In particular, label identification module 1016 and mapping module 1232 may receive the reference token value "1111 … 11" and identify the set of privacy labels 1012 having the actual values "8A8A … 8A", "8B8B … 8B", and "8C8C … 8C". For purposes of authorization, however, the reference token value "1111 … 11" provisioned to user 1001 may be mapped to the reference privacy label value "2222 … 22". At step 1314, the privacy label value "2222 … 22" may be updated by generating a map of actual values "8A8A … 8A", "8B8B … 8B", and "8C8C … 8C" to reference values "7A7A … 7A", "7B7B … 7B", and "7C7C … 7C".

[000116] At a step 1316, authorization module 1020 may generate a set of corresponding permissions 1023 for each privacy label in the set as described above with reference to step 1106 shown in Figure 8. For each privacy label 1012 in the set of privacy labels 1012, authorization module 1020 may identify a corresponding role and generate the set of permissions 1023. Depending on the desired scope of access for user 1001, the set of permissions 1023 may include, for example, a read permission, a write permission, an update permission, a delete permission, and/or a purge permission. The set of permissions 1023 may also define the user as a person, a system, or a proxy for a person or system. For example, for the set of privacy labels 1012 having the actual values "8A8A … 8A", "8B8B … 8B", and "8C8C … 8C", authorization module 1020 may generate a set of corresponding permissions allowing write and delete access but denying purge access for "8A8A … 8A", a set of corresponding permissions only allowing read access for "8B8B … 8B", and a set of corresponding permissions allowing full read, write, update, delete, and purge access for “8C8C … 8C”. In this way, user 1001 may be associated with an updated set of permissions for purposes of making data requests to resources associated with each of the privacy labels 1012 in the set identified by label identification module 1016.

[000117] At a step 1318, authorization module 1020 may provide each map and its corresponding set of permissions 1023 to a resources 1050 corresponding to each respective privacy label in a communication 1227. For example, authorization module 1020 may send a communication 1127a to a resource 1050a. Communication 1127a may include the map of “8A8A … 8A” to “7A7A … 7A” as well as the corresponding set of permissions allowing write and delete access but denying purge access for user 1001. Authorization module 1020 may also send a communication 1127b to a resource 1050b. Communication 1127b may include the map of “8B8B … 8B” to “7B7B … 7B” as well as the corresponding set of permissions only allowing read access for user 1001. Authorization module 1020 may also send a communication 1127c to a resource 1050b. Communication 1127b may include the map of “8C8C … 8C” to “7C7C … 7C” as well as the corresponding set of permissions allowing full read, write, update, delete, and purge access for user 1001. In some examples, authorization module 1020 may regenerate a particular set of permissions 1023 and provide it to the resource 1050. For example, authorization module 1020 may de-provision a role or reassign a role to a user 1001. Authorization module 1020 may regenerate the set of permissions 1023 and provide it to resource 1050 in order to reflect the change. For example, authorization module 1020 may perform a regeneration periodically (e.g., using a timer), upon a triggering event (e.g., a data request), etc.

[000118] At a step 1320, authorization module 1020 may provide a response to authorization request 1024. Response 1024 may include the set of reference privacy labels generated at step 1314. For example, user 1001 may be sent a set of reference privacy labels having token values “7A7A … 7A”, “7B7B … 7B”, and “7C7C … 7C” corresponding to resources 1050a, 1050b, and 1050c respectively. Response 1024 may also include, for example, a resource locator for each resource 1050 corresponding to a privacy label 1012 in the set of privacy labels 1012 for which a set of permissions 1023 was generated. User 1001 may then send data requests to each respective resource 1050 using the corresponding reference privacy label and resource locator. For example, user 1001 may communicate with resource 1050a using the reference token value “7A7A … 7A”, with resource 1050b using the reference token value “7B7B … 7B”, and with resource 1050c using the reference token value “7C7C … 7C”.

[000119] At a step 1322, a communication 1240 from user 1001 may be received by a data broker module 1260. Communication 1240 may include user identifier data 1242 and data request 1244. In some examples, user identifier data 1242 may be out of band data with respect to data request 1244. User identifier data 1242 may include, for example, a reference privacy label provided to user 1001 at step 1320. Data request 1244 may include, for example, a select statement or other type of query that defines the scope of a data request. For example, user 1001 may send a communication 1240 to resource 1050c via a data broker module 1260 using a resource locator received at step 1320. Communication 1240 may include the reference label value “7C7C … 7C” received at step 1320 as user identifier data 1242. Communication 1240 may also include a data request 1244 requesting all emails from person X having Y in the subject line. In some examples, an authorization token and/or a session token may be included in user identifier data 1242 as well.

[000120] At a step 1324, data broker module 1260 may determine whether to validate data request 1244 based on user identifier data 1244. For example, data broker module 1260 may communicate with authorization module 1020, mapping module 1232, and session module 1236 in order to implement a validation process similar to that described with respect to step 1310. If data broker module 1260 determines that data request 1244 is not validated, then at a step 1326, user 1001 may be sent a response to data request 1024 indicating that access is denied. If data broker module 1260 determines that data request 1244 is validated, then at a step 1328, data broker module 1260 may send a communication to a resource 1050 including user identifier data 1242 and data request 1244. For example, data broker module 1260 may send a communication to resource 1050c including the reference label value

"7C7C … 7C" as user identifier data 1242 and the data request 1244 requesting all emails from person X having Y in the subject line.

[000121] At a step 1330, a resource 1050 may receive the communication from data broker module 1260 and process the request in a manner similar to that described above with reference to Figure 4 above. In particular, resource 1050 may, at a step 1332, use the reference value included as user identifier data 1242 and the map provided at step 1318 to identify the corresponding actual privacy label 1012. For example, resource 1050c may receive the communication from data broker module 1260 and use the map provided at step 1318 to identify the actual privacy label value "8C8C … 8C" corresponding to the reference value "7C7C … 7C".

[000122] At a step 1332, resource 1050 may additionally determine whether user 1001 has a permission required for data request 1244 using the corresponding set of permissions for the privacy label received at step 1318. For example, resource 1050c may determine that, based on reference value "7C7C … 7C" and the mapped actual privacy label value "8C8C … 8C", user 1001 has a corresponding set of permissions allowing full read, write, update, delete, and purge access. As such, resource 1050c may process data request 1244. At a step 1334, resource 1050 may send user 1001 a response 1246 to data request 1244 including the requested data. The response may be sent via data broker module 1260.

[000123] The examples of apparatus, methods, and systems for multiple data storage and separation, authentication, and authorization disclosed herein may be variously combined to facilitate functionality heretofore untenable in multi-tenancy systems that use multiple servers and databases for each client or resource, with each server hosting its own authentication and authorization systems. One example of such functionality is data sandboxing. The term "data sandbox" as used herein refers to a secure computing environment in which a set of data may be temporarily isolated, stored and analyzed, and in which access to the data is limited to a specific set of authorized users. The term "sandboxing" generally refers to the various acts of generating, configuring, using, and/or removing a data sandbox. Data sandboxing may be desired where, for example, a user has need to analyze data sets from multiple different entities and/or resources, but may not have the necessary permissions to access records for all of the entities and/or resources. The user may desire to create a data sandbox accessible by only persons or systems having the requisite permissions in which to temporarily place comingled sets of data for analysis.

[000124] For example, a provider of cloud-based services, using the various examples of apparatus, methods, and systems for multiple data storage and separation, authentication, and authorization disclosed herein, may deploy a single server running an operating system and hosting multiple instances of a resource for multiple different users (e.g., clients or other entities) with a single database storing records for each of the users. The system may provide centralized authorization and authentication per the examples described herein, with granular access control for each of the users. The service provider may wish to provide data sandboxing functionality for data sets that include comingled data for the multiple different users, where such users may have differing access permissions in place based on the sensitivity of each of their respective data sets. Such differing permissions may include, for example, privacy labels that define exclusive data sets accessible by some users, but not others.

[000125] Figure 14 is a schematic illustration of an example data sandboxing apparatus 1400 for use with a multiple user data storage and separation apparatus, such as apparatus 100 shown in Figure 1. Apparatus 1400 may be, for example, a shared data sandboxing component of a cloud computing system used to provide multiple different business resources, such as email applications and case management systems, to multiple different users 1401. As will be described hereafter, in general, apparatus 1400 may include data sandboxing functionality.

[000126] Apparatus 1400 may include processing electronics 1402. Processing electronics 1402 may be similar to processing electronics 102 as shown in and described with reference to Figure 1. For example, as shown in Figure 14, processing electronics 1402 may include a processor 1404 configured to execute logic in the form of instruction modules contained in a memory 1406, similar to processor 104 and memory 106 as shown in and described with reference to Figure 1. As with processor 104 and memory 106, processor 1404 and memory 1406 may be single devices, or may comprise multiple processors and memories having distributed functionality.

[000127] Memory 1406 may include records 1408 stored therein. Records 1408 may be similar to records 108 as shown in and described with reference to Figure 1. For example, records 1408 may include user data items for multiple different users and may be defined by and stored using correlated tree structures 1410 similar to tree structures 110 as shown in and described with reference to Figure 1. In particular, tree structures 1410 may have schema definitions such that stored records 1408 are unrestricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns.

[000128] Memory 1406 may also include privacy labels 1412 stored therein. Privacy labels 1412 may be similar to privacy labels 112 as shown in and described with reference to Figure 1. For example, privacy labels 1412 may distinguish among multiple different users 1401 and/or may define exclusive groupings of records 1408 for the multiple different users. Each privacy label 1412 may include a unique identification mapping to a user 1401. In some examples, privacy label 1412 may include a number of tokens (e.g., unique randomly generated cryptographically entropic values having a predetermined number of bytes, etc.) that are uniquely provisioned to user 1401. Privacy labels 1412 may be correlated with each of records 1408 using privacy label tree structures included in tree structures 1410. The privacy label tree structures defining privacy labels 1412 may have schema definitions similar to other tree structures 1410. That is, privacy labels 1412 may be structured and stored without being restricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns. In some examples, records 1408 and privacy labels 1412 may be stored and maintained in a remote and/or separate multiple user data storage and separation system.

[000129] Memory 1406 may also include sandbox configuration module 1414, a data request module 1415, a correlation module 1416, and a data removal module 1417. Data request module 1415, correlation module 1416, and data removal module 1417 may be included in a data sandbox 1418 generated by module 1414 along with a local memory 1419. Modules 1414, 1415, 1416, and 1417 may cooperate to cause processing electronics 1402 to carry out the process 1500 set forth by the flow diagram of Figure 15. As indicated by a step 1502, sandbox configuration module 1414 may generate data sandbox 1418.

[000130] In some examples, sandbox configuration module 1414 may generate data sandbox 1418 by generating a virtual machine including the computing environment for data sandbox 1418. The term "virtual machine" as used herein refers to a computing environment that emulates a complete host computer and on which a guest operating system may boot and run as on actual hardware, but may only access host resources through the emulator. In some examples, a suitable virtual machine may already be in place to host data sandbox 1418. In some examples, sandbox configuration module 1414 may utilize an autonomous provisioning module to automatically provision an initial privacy label to data sandbox 1418 and to instruct data sandbox 1418 to communicate with sandbox configuration module 1414 using the privacy label in order to be configured.

[000131] In some examples, sandbox configuration module 1414 may generate data sandbox 1418 in response to a data sandbox configuration request 1422 from user 1401. A data sandbox configuration request may be, for example, a manual request submitted via a user interface provided by sandbox configuration module 1414 (e.g., user 1401 is a person). A data sandbox configuration request 1422 may also be, for example, an automated request (e.g., user 1401 is a computing device hosting an automated process for triggering a data sandbox configuration request 1422). A data sandbox configuration request 1422 may include, for example, data indicating the scope of a data query to be used by a data sandbox to identify a particular set of records 1408 for analysis. In particular, the request may include exclusive groupings of records 1408 for the multiple different users as defined by privacy labels 1412. A data sandbox configuration request 1422 may also include, for example, data indicating the particular type of analysis requested, such as a particular correlation process to be used to identify relationships among records 1408 accessed by data sandbox 1418.

[000132] Sandbox configuration module 1414 may configure data sandbox 1418 with a query that may be used by data sandbox 1418 to identify a particular set of records 1408 for analysis. Sandbox configuration module 1414 may also configure data sandbox 1418 with credentials that may be used to authenticate data sandbox 1418. Sandbox configuration module 1414 may also configure data sandbox 1418 by provisioning a number of privacy labels 1412 to data sandbox 1418 having associated permission sets that allow data sandbox 1418 to access and temporarily store exclusive groupings of records 1408 for the multiple different users that may be included in the set of records identified by the query. Sandbox configuration module 1414 may also configure data sandbox 1418 to limit access to records 1408 stored in local memory 1419. In some examples, access to local memory 1419 may be limited to a particular set of users having the requisite permissions to access the full set of records 1408 stored in memory 1418 (e.g., persons or computing devices that have been provisioned with privacy labels 1412 corresponding to each exclusive set of data stored in local memory 1419). In some examples, access to local memory may be limited to only data sandbox 1418.

[000133] At a step 1504, data request module 1415 may access a comingled set of records 1408 selected from among exclusive groupings of records 1408. For example, a query used by data request module 1415 to identify a particular set of records 1408 for analysis may include exclusive groupings of records 1408 for multiple different users. As such, the comingled set of records may require different sets of permissions and/or privacy labels 1412. Data sandbox 1418 may be authenticated and authorized to access the comingled set of records 1408 using credentials, privacy labels, and permission sets configured by sandbox configuration module 1414. Data sandbox 1418 may then use the query to identify and access the comingled set of records. At a step 1506, data sandbox 1418 may store the comingled set of records 1408 in local memory 1419.

[000134] At a step 1508, correlation module 1416 may execute a correlation process to identify relationships among records 1408 in the comingled set of records 1408. For example, the correlation process may examine segments associated with each of the records to identify the relationships. In some examples, the partial content of segments are examined (e.g., text within comments). In some examples, dissimilar segment types may be examined to identify unknown relationships among records 1408. In some examples, the noted relationships may be stored and included in a report referencing the identity and location of correlated records and an indication of the degree of correlation between records in the comingled set. In some examples the report may be escalated to users having the requisite permissions to access records stored in local memory 1419.

[000135] At a step 1510, the comingled set of records 1408 may be deleted from local memory 1419. In some examples, the comingled set of records may be deleted upon the occurrence of a predetermined event. For example, comingled set of records 1408, may be automatically deleted from local memory 1419 after a predetermined period of time after they are stored. In some examples, comingled set of records 1408 may be deleted upon completion of a review by a user having the requisite permissions to access records stored in local memory 1419.

[000136] Figure 16 is a schematic illustration of another example data sandboxing apparatus 1600 for use with a multiple user data storage and separation apparatus, such as apparatus 100 shown in Figure 1. As will be described hereafter, apparatus 1600 may include data sandboxing functionality. Apparatus 1600 is similar to apparatus 1400 shown in and described with reference to Figure 14. Those components of apparatus 1600 that correspond to apparatus 1400 are numbered similarly.

[000137] Apparatus 1600 may provide sandboxing functionality in the context of a single server running an operating system 1603. The server may host, for example, a capability-based system including a single centralized database storing records 1408 for multiple different users. The server may provide centralized authorization and authentication at the application layer with granular access control per the examples described herein using an authentication module 1610 and an authorization module 1620. The server may also host multiple instances 1650 of a resource for multiple different users (e.g., clients or other entities). For example, instances 1650a (I1), 1650b (I2), and 1650n (IN) may each be an instance of a case management service provided by a cloud-based service provider and respectively corresponding to each of N different clients. Case management tickets may be created, edited, updated, and reported for each client and stored as records 1408 in the centralized database.

[000138] The service provider may wish to provide data sandboxing functionality for data sets that include comingled data for the multiple different clients, where such clients may have differing access permissions in place based on the sensitivity of each of their respective data sets. In particular, privacy labels 1410 may be used to distinguish among multiple different users 1401 and/or may define exclusive groupings of records 1408 for the multiple different users. For example, a user 1401a may be an analyst that is able to access records 1408 for a client associated with instance 1650a, but not for instances 1650b – 1650n. Other users 1401 may be analysts respectively assigned to and able to access records 1408 for each of instances 1650b – 1650n (but only for their respective instance). In contrast, user 1401b may be a manager for the cloud-based service provider who may access records 1408 associated with each of instances 1650b – 1650n.

[000139] Modules 1414, 1415, 1416, and 1417 may cooperate to cause processing electronics 1402 to carry out the process 1700 set forth by the flow diagram of Figure 17. As indicated by a step 1702, sandbox configuration module 1414 may receive a data sandbox configuration request 1422 from a user 1401. A data sandbox configuration request may be, for example, a manual request submitted via a user interface provided by sandbox configuration module 1414 (e.g., user 1401 is a person). A data sandbox configuration request 1422 may also be, for example, an automated request (e.g., user 1401 is a computing device hosting an automated process for triggering a data sandbox configuration request 1422). For example, user 1401a may review a ticket from the case management system hosted by instance 1550a and note that the server hosting each of the case management systems may be experiencing performance issues impacting instance 1650a. User 1401a may desire to investigate further to determine if a larger problem exists that may warrant a higher priority. User 1401a may access a user interface provided by sandbox configuration module 1414 and enter a data sandbox configuration request 1422.

[000140] A data sandbox configuration request 1422 may include, for example, data indicating the scope of a data query to be used by a data sandbox to identify a particular set of records 1408 for analysis. In particular, the request may include exclusive groupings of records 1408 for the multiple different users as defined by privacy labels 1412. A data sandbox configuration request 1422 may also include, for example, data indicating the particular type of analysis requested, such as a particular correlation process to be used to identify relationships among records 1408 accessed by data sandbox 1418. For example, user 1401a may specify particular types of information that may be contained in segments of records 1408 which may be if interest. User 1401a may also specify which clients or instances may have associated records 1408 that may contain information useful in diagnosing a problem with the server.

[000141] At a step 1704, sandbox configuration module 1414 may generate a virtual machine including the computing environment for a data sandbox 1418. In some examples, a suitable virtual machine may already be in place. In some examples, sandbox configuration module 1414 may utilize an autonomous provisioning module 1680 to automatically provision an initial privacy label to data sandbox 1418 and to instruct data sandbox 1418 to communicate with sandbox configuration module 1414 using the privacy label in order to be configured.

[000142] At a step 1706, sandbox configuration module 1414 may configure data sandbox 1418. In particular, sandbox configuration module may configure data sandbox 1418 with a query that may be used by data sandbox 1418 to identify a particular set of records 1408 for analysis. Sandbox configuration module 1414 may also configure data sandbox 1418 with credentials that may be used by authentication module 1410 to authenticate data sandbox 1418. Sandbox configuration module 1414 may also configure data sandbox 1418 by provisioning a number of privacy labels 1412 to data sandbox 1418 having associated permission sets (e.g., “System” and “Read”) that allow data sandbox 1418 to be authorized by authorization module 1520, and then access and temporarily store exclusive groupings of records 1408 for the multiple different users that may be included in the set of records identified by the query. Sandbox configuration module 1414 may also configure data sandbox 1418 to limit access to records 1408 stored in local memory 1419. In some examples, access to local memory 1419 may be limited to a particular set of users having the requisite permissions to access the full set of records 1408 stored in memory 1418 (e.g., persons or computing devices that have been provisioned with privacy labels 1412 corresponding to each exclusive set of data stored in local memory 1419). In some examples, access to local memory may be limited to only data sandbox 1418.

[000143] At a step 1708, data request module 1415 may access a comingled set of records 1408 selected from among exclusive groupings of records 1408. For example, a query used by data request module 1415 to identify a particular set of records 1408 for analysis may include exclusive groupings of records 1408 for multiple different users. As such, the comingled set of records may require different sets of permissions and/or privacy labels 1412. Data sandbox 1418 may be authenticated and authorized to access the comingled set of records 1408 using credentials, privacy labels, and permission sets configured by sandbox configuration module 1414. Data sandbox 1418 may then use the query to identify and access the comingled set of records. Data sandbox 1418 may store the comingled set of records 1408 in local memory 1419.

[000144] At a step 1710, correlation module 1416 may execute a correlation process to identify relationships among records 1408 in the comingled set of records 1408. For example, the correlation process may examine segments associated with each of the records to identify the relationships. In some examples, the partial content of segments are examined (e.g., text within comments). In some examples, dissimilar segment types may be examined to identify unknown relationships among records 1408. At a step 1712, the noted relationships may be stored and included in a report referencing the identity and location of correlated records and an indication of the degree of correlation between records in the comingled set. For example, correlation module 1416 may note in a report 1426 that particular records for instance 1650b include segments correlated with a record associated with instance 1650a. The segments may be of dissimilar types, but may be correlated due to the inclusion of the IP address for the server, the term “Performance”, etc. Report 1426 may be provided to user 1401a by a reporting module 1690.

[000145] At a step 1714, reporting module 1690 may escalate report 1426 to a user having the requisite permissions to access records 1408 identified in report 1426 and stored in local memory 1419. For example, reporting module 1690 may determine that user 1401a does not have the required permissions to access records for instance 1650b. Reporting module 1690 may escalate report 1426 to user 1401b for review. User 1401b may then access the records stored in local memory 1419 if desired.

[000146] At a step 1716, the comingled set of records 1408 may be deleted from local memory 1419. In some examples, the comingled set of records may be deleted upon the occurrence of a predetermined event. For example, comingled set of records 1408, may be automatically deleted from local memory 1419 after a predetermined period of time after they are stored. In some examples, comingled set of records 1408 may be deleted upon completion of a review by a user having the requisite permissions to access records 1408 stored in local memory 1419. For example, user 1401b may provide an indication to record removal module 1519 that a review has been completed. Record removal module 1619 may then delete records 1408 from local memory 1419. In some examples, at a step 1718, sandbox configuration module 1414 may remove the virtual machine hosting data sandbox 1418 in order to remove it from the system.

[000147] Another example of how the apparatus, methods, and systems for multiple data storage and separation, authentication, and authorization disclosed herein may be variously combined to facilitate functionality heretofore untenable in multi-tenancy systems is resource provisioning. In particular, the resource provisioning examples disclosed herein may separate management of a host infrastructure at the operating system level from management of each hosted resource instance at the application layer while maintaining appropriate data separation and storage. Such examples may facilitate automated near real-time deployment of resources. The term "provisioning" as used herein refers to providing a user with an allocation and/or assignment of information for accessing data and/or resources in a computing system. For example, in a capability-based system in accordance with the examples described herein, users (e.g., persons, system components, etc.) may be provisioned a number of privacy labels that may be used by the system to grant or deny the users access to data or resources. Similarly, a user may be provisioned a number of resource locators defining where and how particular data and resources may be accessed. Where the user is an actual resource, such as a newly deployed business resource, the resource may be provisioned a privacy label upon initial configuration. The privacy label may be, for example, a number of tokens that identify the resource to other system components and allow it to access such components. The privacy label may also be a primary and/or root privacy label for purposes of storing records associated with the resource, and may serve as the basis for a hierarchy of privacy labels for users as defined by a privacy label tree structure. A user is "provisioned" when information for accessing data and/or resources in a computing system, such as a privacy label, has been provided to the user.

[000148] A provider of cloud-based services, using the various examples of apparatus, methods, and systems for multiple data storage and separation, authentication, and authorization disclosed herein, may deploy a single server running an operating system capable of hosting multiple instances of a resource for multiple different users (e.g., clients or other entities) with a single database storing records for each of the users. The system may provide centralized authorization and authentication per the examples described herein, with granular access control for each of the users. The service provider may wish to provide infrastructure (e.g., operating system level) management for multiple instances of the resource hosted by the server for multiple different clients, while at the same time allowing an alternate provider to provide instance management for each of the multiple instances. The service provider may desire to facilitate near real time deployment of each instance for the alternate provider, while at the same time maintaining constraints on the scale of the service (e.g., limiting the number of instances that may be requested and managed by the alternate provider).

[000149] Figure 18 is a schematic illustration of an example resource provisioning apparatus 1800 for use with a multiple user data storage and separation apparatus, such as apparatus 100 shown in Figure 1. Apparatus 1800 may be, for example, a shared resource provisioning component of a cloud computing system used to provide multiple different business resources, such as email applications and case management systems, to multiple different users. As will be described hereafter, in general, apparatus 1800 may include resource provisioning functionality. In particular, apparatus 1800 may facilitate separation of infrastructure management and instance management of a resource, and/or may facilitate timely and efficient deployment of resource instances for multiple different users (e.g., clients, customers, etc.).

[000150] Apparatus 1800 may include processing electronics 1802. Processing electronics 1802 may be similar to processing electronics 102 as shown in and described with reference to Figure 1. For example, as shown in Figure 18, processing electronics 1802 may include a processor 1804 configured to execute logic in the form of instruction modules contained in a memory 1806, similar to processor 104 and memory 106 as shown in and described with reference to Figure 1. As with processor 104 and memory 106, processor 1804 and memory 1806 may be single devices, or may comprise multiple processors and memories having distributed functionality.

[000151] Memory 1806 may include schema for storing records 1808 therein. Records 1808 may be similar to records 108 as shown in and described with reference to Figure 1. For example, records 1808 may include user data items for multiple different users and may be defined by and stored using correlated tree structures 1810 similar to tree structures 110 as shown in and described with reference to Figure 1. In particular, tree structures 1810 may have schema definitions such that stored records 1808 are unrestricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns.

[000152] Memory 1806 may also include schema for storing privacy labels 1812 therein. Privacy labels 1812 may be similar to privacy labels 112 as shown in and described with reference to Figure 1. For example, privacy labels 1812 may distinguish among multiple different users and/or may define exclusive groupings of records 1808 for the multiple different users. Each privacy label 1812 may include a unique identification mapping to a user. In some examples, privacy label 1812 may include a number of tokens (e.g., unique randomly generated cryptographically entropic values having a predetermined number of bytes, etc.) that are uniquely provisioned to a user. Privacy labels 1812 may be correlated with each of records 1808 using privacy label tree structures included in tree structures 1810. The privacy label tree structures defining privacy labels 1812 may have schema definitions similar to other tree structures 1810. That is, privacy labels 1812 may be structured and stored without being restricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns. In some examples, records 1808 and privacy labels 1812 may be stored and maintained in a remote and/or separate multiple user data storage and separation system.

[000153] Memory 1806 may also include a resource template 1814. Resource template 1814 may be, for example, a template structure that may be used to host components of any of multiple different resources, such as an email application, a scheduler, a case management system, etc. In some examples, resource template 1814 may be contained in a virtual machine compartment that may be configured as a particular resource by provisioning information, such as a privacy label, to resource template 1814 and then uploading any instruction modules (e.g., executable files, etc.) needed to configure resource template 1814 for its intended purpose. Resource template 1814 may include basic modules for communicating configuration requests to other system components.

[000154] Memory 1816 may also include a provisioning module 1816. In some examples, resource template 1814 and provisioning module 1816 may be provided by the same processing electronics 1802. In some examples, resource template 1814 and provisioning module 1816 may be provided by distinct and/or separate systems or apparatus having separate or distinct locations and/or processing electronics. Provisioning module 1816 may cause processing electronics 1802 to carry out the process 1900 set forth by the flow diagram of Figure 19. As indicated by a step 1902, provisioning module 1816 may receive a configuration request 1822 from resource template 1814. For example, resource provisioning apparatus 1800 may receive a request to provide a resource template 1814 for a new scheduling application to be deployed. In response, resource provisioning apparatus 1800 may activate a virtual machine containing resource template 1814. Upon activation of the virtual machine, resource template 1814 may send configuration request 1822. In some examples, a suitable virtual machine may already be in place.

[000155] Resource template 1814 may send configuration request 1822 to a preconfigured location or home. In some examples, resource template 1814 may contain a preconfigured resource locator for an infrastructure manager and an instruction to send configuration request 1822 to the infrastructure manager using the preconfigured resource locator. The infrastructure manager may be, for example, a service provider (and/or an automated computing system associated therewith) that may provide infrastructure (e.g., operating system level) management for multiple instances of a resource hosted by a server, while at the same time allowing an alternate provider to provide instance management (e.g., application layer management) for each of the multiple instances. For example, the infrastructure manager may maintain a server hosting parallel instances of a resource contained in a virtual machine compartment and provide, for example, backup services and patching services. In some examples, resource provisioning apparatus 1800 may be provided by the same entity providing infrastructure management. In some examples, resource provisioning apparatus 1800 and infrastructure management may be provided by different entities. An instance manager 1824 may, for example, configure and/or customize each resource template 1814 at the application layer and provide an interface for adding users, defining roles and permissions for each user, assigning privacy labels to users, reading, writing, updating, deleting, and purging records, etc. Instance manager 1824 may be, for example, an automated computing system provided by an instance management entity and containing processing electronics that allow it to respond to configuration requests.

[000156] At a step 1904, provisioning module 1816 may provision a privacy label 1812 to resource template 1814. In some examples, privacy label 1812 may be, for example, a number of tokens that identify resource template 1814 to other system components and allow it to access such components. Privacy label 1812 may also be an initial, primary and/or root privacy label for purposes of storing records 1808 associated with a resource once resource template 1814 has received primary function configuration (e.g., from an instance manager) and users have been added and have been provisioned with appropriate access permissions and privacy labels. For example, initial privacy label 1812 may serve as the basis for a hierarchy of privacy labels 1812 for users as defined by a privacy label tree structure 1810. Provisioning module 1816 may also provision a resource locator 1818 to resource template 1814. The resource locator 1818 may be, for example, an IP address or other information on how to communicate with an instance manager.

[000157] At a step 1906, provisioning module 1816 may respond to communication request 1822 by sending privacy label 1812 to resource template 1814. Provisioning module 1816 may also send an instruction to resource template 1814 to send a configuration request 1826 to an instance manager 1824. In some examples, request 1826 may include privacy label 1812 and a request for an application layer configuration to establish the primary function of resource template 1814. For example, provisioning module 1816 may instruct resource template 1814 to send a communication 1826 to instance manager 1824 requesting an application layer configuration 1828 to configure resource template 1814 as an email application, a scheduler, a case management system, etc. Instance manager 1824 may respond to the configuration request by sending application layer configuration 1828. Application layer configuration 1828 may include, for example, any instruction modules (e.g., executable files, etc.) needed to configure resource template 1814 for its intended purpose. Upon loading application layer configuration 1828, resource template 1814 may be fully enabled as a resource, and instance manager 1824 may notify a user that the resource has been configured and is ready to use. Instance manager 1824 may provide the user with an interface for adding users, defining roles and permissions for each user (e.g., reading, writing, updating, deleting, and purging records, etc.), assigning privacy labels to users, etc. Privacy label 1812 may be an initial, primary and/or root privacy label for purposes of storing records 1808 associated with the resource, and may serve as the basis for a hierarchy of privacy labels 1812 for users as defined by a privacy label tree structure 1810.

[000158] Figure 20 is a flow diagram of an example resource provisioning process 2000 that may be carried out by resource template 1814 shown in Figure 18. At a step 2002, resource template 1814 may send a configuration request 1822 to provisioning module 1816. For example, resource provisioning apparatus 1800 may receive a request to provide a resource template 1814 for a new scheduling application to be deployed. In response, resource provisioning apparatus 1800 may activate a virtual machine containing resource template 1814. Upon activation of the virtual machine, resource template 1814 may send configuration request 1822. In some examples, a suitable virtual machine containing resource template 1814 may already be in place. Resource template 1814 may send configuration request 1822 to a preconfigured location or home. In some examples, resource template 1814 may contain a preconfigured resource locator for an infrastructure manager and an instruction to send configuration request 1822 to the infrastructure manager using the preconfigured resource locator.

[000159] At a step 2004, resource template 1814 may receive a response from provisioning module 1816 including a privacy label 1812 provisioned by provisioning module 1816. In some examples, resource template 1814 may also receive a resource locator 1818 provisioned by provisioning module 1816. Resource template 1814 may also receive an instruction from provisioning module 1816 to send a configuration request 1826 to an instance manager 1824. In some examples, request 1826 may include privacy label 1812 and a request for an application layer configuration to establish the primary function of resource template 1814. For example, provisioning module 1816 may instruct resource template 1814 to send a communication 1826 to instance manager 1824 requesting an application layer configuration 1828 to configure resource template 1814 as an email application, a scheduler, a case management system, etc. At a step 2006, resource template 1814 may send configuration request 1826 to instance manager 1824.

[000160] At a step 2008, resource template 1814 may receive application layer configuration 1828 from instance manager 1824 in response to configuration request 1826. Application layer configuration 1828 may include, for example, any instruction modules (e.g., executable files, etc.) needed to configure resource template 1814 for its intended purpose. Upon loading application layer configuration 1828, resource template 1814 may be fully enabled as a resource, and instance manager 1824 may notify a user that the resource has been configured and is ready to use. Instance manager 1824 may provide the user with an interface for adding users, defining roles and permissions for each user, assigning privacy labels to users, reading, writing, updating, deleting, and purging records, etc. Privacy label 1812 may be an initial, primary and/or root privacy label for purposes of storing records 1808 associated with the resource, and may serve as the basis for a hierarchy of privacy labels 1812 for users as defined by a privacy label tree structure 1810.

[000161] Figure 21 is a schematic illustration of another example resource provisioning apparatus 2100 for use with a multiple user data storage and separation apparatus, such as apparatus 100 shown in Figure 1. As will be described hereafter, in general, apparatus 2100 may include resource provisioning functionality. In particular, apparatus 2100 may facilitate separation of infrastructure management and instance management of a resource, and/or may facilitate timely and efficient deployment of resource instances for multiple different users (e.g., clients, customers, etc.). Apparatus 2100 is similar to apparatus 1800 shown in and described with reference to Figure 18. Those components of apparatus 2100 that correspond to apparatus 1800 are numbered similarly.

[000162] Apparatus 2100 may provide resource provisioning in the context of a single server running an operating system 2103. The server may host, for example, a capability-based system including a single centralized database having schema for storing records 1808 for multiple different users. The server may provide centralized authorization and authentication at the application layer with granular access control per the examples described herein using an authentication module 2110 and an authorization module 2120. The server may also host multiple resource templates 1814 that may be configured as resource instances for multiple different users (e.g., clients or other entities). For example, resource templates 1814a (I1), 1814b (I2), and 1814n (IN) may each be contained in a respective compartment of a virtual machine, and may be converted into, for example, instances of a scheduling resource a case management service provided by a cloud-based service provider and respectively corresponding to each of N different clients 2130 (e.g., clients 2130a, 2130b, …2130n). Scheduled appointments may be created, edited, updated, and reported for each client and stored as records 1808 in the centralized database.

[000163] Apparatus 2100 may provide autonomous resource provisioning functionality. For example, a service provider may wish to provide infrastructure (e.g., operating system level) management for multiple instances of the scheduling resource that may be hosted by the server for multiple different clients, while at the same time allowing an alternate provider to provide instance management for each of the multiple instances of the scheduling resource. The service provider may desire to facilitate near real time deployment of each instance of the resource for the alternate provider, while at the same time maintaining constraints on the scale of the service (e.g., limiting the number of instances that may be requested and managed by the alternate provider). To facilitate such autonomous resource provisioning, apparatus 2100 may include an infrastructure manager 2140. In some examples, infrastructure manager 2140 may be hosted outside of apparatus 2100 (e.g., hosted on another server, hosted by a different entity, etc.). Infrastructure manager 2140 may include an activation module 2142 and provisioning module 1816. Infrastructure manager 2140 may also include operating system configurations 2144 that may be used to configure resource templates 1814 for use within apparatus 2100.

[000164] Modules 1816 and 2142 may cooperate to cause processing electronics 1802 to carry out the process 2200 set forth by the flow diagram of Figure 22. As indicated by a step 2202, activation module 2142 may receive an activation request 2152 requesting activation of a resource template 1814. For example, a client 2310a may request to purchase (e.g., via an online request) an instance of a scheduling resource managed by a service provider hosting infrastructure manager 2140. In response to the request, the service provider, via infrastructure manager 2140, may confirm the request and initiate procurement of the scheduling resource.

[000165] At a step 2204, activation module 2142 may activate a resource template 1814 that may be configured as an instance of the scheduling resource. In some examples, activation module 2140 may need to activate a virtual machine containing resource template 1814. In some examples, a suitable virtual machine containing resource template 1814 may already be in place. Upon activation, resource template 1814 may send configuration request 1822. Resource template 1814 may send configuration request 1822 to a preconfigured location or home. In some examples, resource template 1814 may contain a preconfigured resource locator for infrastructure manager 2140 and an instruction to send configuration request 1822 to infrastructure manager 2140 using the preconfigured resource locator. At a step 2206, provisioning module 1816 may receive configuration request 1822 from resource template 1814.

[000166] At a step 2208, provisioning module 1816 may provision a privacy label 1812 to resource template 1814. In some examples, privacy label 1812 may be, for example, a number of tokens that identify resource template 1814 to other system components and allow it to access such components. Privacy label 1812 may also be an initial, primary and/or root privacy label for purposes of storing records 1808 associated with a resource once resource template 1814 has received primary function configuration (e.g., from an instance manager) and users have been added and have been provisioned with appropriate access permissions and privacy labels. For example, initial privacy label 1812 may serve as the basis for a hierarchy of privacy labels 1812 for users as defined by a privacy label tree structure 1810. Provisioning module 1816 may also provision a resource locator 1818 to resource template 1814.

The resource locator 1818 may be, for example, an IP address or other information on how to communicate with instance manager 1824.

[000167] At a step 2210, provisioning module 1816 may respond to communication request 1822 by sending privacy label 1812 to resource template 1814. Provisioning module 1816 may also hand off instance management of resource template 1814 by sending an instruction to resource template 1814 to send a configuration request 1826 to instance manager 1824. In some examples, request 1826 may include privacy label 1812 and a request for an application layer configuration to establish the primary function of resource template 1814. For example, provisioning module 1816 may instruct resource template 1814 to send a communication 1826 to instance manager 1824 requesting an application layer configuration 1828 to configure resource template 1814 as the scheduling resource purchased by client 2130a. Resource template 1814 may respond to the instruction by sending configuration request 1826 to instance manager 1824, and at a step 2212, instance manager 1824 may receive configuration request 1826.

[000168] At a step 2214, instance manager 1824 may respond to configuration request 1826 by sending application layer configuration 1828. Application layer configuration 1828 may include, for example, any instruction modules (e.g., executable files, etc.) needed to configure resource template 1814 for its primary function and/or intended purpose. Upon loading application layer configuration 1828, resource template 1814 may be fully enabled as a resource. For example, application layer configuration 1828 may include executable instruction modules that, when executed, configure resource template 1814 as the scheduling resource purchased by client 2130a.

[000169] At a step 2216, infrastructure manager 2140 may receive a confirmation message from resource template 1814. For example, resource template 1814 may upload application layer 1828 and successfully configure the scheduling resource purchased by client 2130a. Resource template 1814 may then send a confirmation message to infrastructure manager 2140 informing it that configuration was successful. At a step 2218, a notification may be sent to a client 2130 informing the user that resource template 1814 has been configured and is ready for use. For example, upon successful configuration of resource template 1814 as the scheduler resource purchased by client 2130a, infrastructure manager 2140 and/or instance manager 1824 may send a notification to client 2130a.

[000170] At a step 2220, instance manager 1824 may receive data 2154 including new user data and role data from client 2130. In some examples, instance manager 1824 may provide a client 2130 with an interface for adding new users 2160, defining roles and permissions for each user 2160 (e.g., reading, writing, updating, deleting, and/or purging records, etc.), assigning privacy labels 1812 to new users 2160, etc. For example, client 2130a may be an administrator for a clinic who wishes to add an employee as a user 2160a of the scheduling resource. Client 2130a may define a number of possible roles (e.g., administrator, doctor, employee, patient, etc.) that may be provisioned to new users 2160. Each role may include a specific set of permissions (e.g., reading, writing, updating, deleting, and/or purging records, etc.) that define a scope of access to the resource and any associated records 1808. The set of permissions may also define the user as a person, a system, or a proxy for a person or system. Client 2130a may provision roles to new users 2160 by correlating each role with privacy labels 1012 that may be provisioned to a user 2160 at a step 2222. For example, client 2130a may assign the role “Employee” to user 2160a by correlating the role with a privacy label 1812 that may be newly provisioned to user 2160a. The role “Employee” may define a set of permissions allowing read-only access to the scheduler resource. Similarly, client 2130a may assign the role “Patient” to a user 2160b. The role “Patient” may define a set of permissions allowing user 2160b to request an appointment. Data 2154 including a listing of roles and users to be correlated with privacy labels 1812 may be sent to instance manager 1824.

[000171] At a step 2222, privacy labels may be provisioned to new users 2160. As set forth above, the privacy label 1812 provisioned to resource template 1814 by provisioning module 1816 may be an initial, primary and/or root privacy label for purposes of storing records 1808 associated with the resource, and may serve as the basis for a hierarchy of privacy labels 1812 for new users 2160 added by client 2160 as defined by a privacy label tree structure 1810. Figure 23 illustrates an example privacy label tree structure 2300 that may be implemented by apparatus 2100. A privacy label 2302 in tree structure 2300 may indicate in initial privacy label provisioned to a resource template 1814 by provisioning module 1816. Additional privacy labels may be provisioned to users using privacy label 2302 as a conceptual root of a hierarchical tree structure. For example, a privacy label 2304 may be provisioned by instance manager 1824 to an administrator for client 2130a, and a privacy label 2306 may be provisioned to an administrator for instance manager 1824. The role "Administrator" may be provisioned by correlating it with each of privacy labels 2304 and 2306 to define a set of permissions for the administrator for client 2130a and the administrator for instance manager 1824. Similarly, a privacy label 2308 may be provisioned by the administrator for client 2130a to user 2160a, an employee of client 2130a. The role "Employee" may be provisioned to the employee 2160a by correlating it with privacy label 2308. The role "Employee" may allow employee 2160a to, for example, review appointments made for patients. Likewise, a privacy label 2308 may be provisioned by the administrator for client 2130a to user 2160b, a patient of client 2130a. The role "Patient" may be provisioned to the patient 2160b by correlating it with privacy label 2308. The role "Patient" may allow patient 2160b to request appointments. Additional privacy labels may be provisioned for new users as desired. As shown in, for example, Figure 4, each privacy label in tree structure 2300 may in turn be correlated with receipts that define records (e.g., records 1808 shown in Figure 21) in order to maintain appropriate data separation.

[000172] As will be understood, the organizational structure reflected by tree structure 2300 may not dictate the actual structure of the privacy labels and/or tokens. In some examples, the privacy labels in tree structure 2300 may be implemented as sets and subsets of tokens that increase or decrease in the number of tokens depending on the level of the privacy label in the hierarchy. Other provisioning schemes for privacy labels are contemplated as well. For example, unique privacy labels having different combinations of a number of bytes may be provisioned regardless of level in the organizational hierarchy rather than provisioning subsets of tokens from a full set of tokens.

[000173] Referring again to Figure 22, at a step 2224, new users 2160 may be notified that the scheduling resource is available for use. For example, client 2160a and/or instance manager 1824 may send each new users 2160a and/or 2160b a message including any privacy labels 1812 provisioned to the user 2160 as well as a resource locator directing the user 2160 to the scheduling resource. In some examples, reference tokens and/or reference labels may be used instead of the actual privacy labels provisioned to the user 2160. In some examples, client software may also be provided in order to facilitate access to the resource. Process 2200 may be repeated to create instances 1814b … 1814n of a scheduler resource for clients 2130b … 2130n. In some examples, infrastructure manager 2140 may place constraints on scale by limiting the number of resource templates 1814 that may be utilized by an instance manager.

[000174] Figure 24 is a schematic illustration of an example workflow management apparatus 2400 for use with a multiple user data storage and separation apparatus, such as apparatus 100 shown in Figure 1. Apparatus 2400 may be, for example, a shared workflow management component of a cloud computing system used to provide multiple different business resources, such as email applications and case management systems, to multiple different users. As will be described hereafter, in general, apparatus 2400 may include workflow management functionality. In particular, apparatus 2400 may allow a user to generate a workflow. The term “workflow” as used herein generally refers to a set of steps defining a sequence of tasks corresponding to functions that may be carried out by a business resource.

[000175] Apparatus 2400 may include processing electronics 2402. Processing electronics 2402 may be similar to processing electronics 102 as shown in and described with reference to Figure 1. For example, as shown in Figure 24, processing electronics 2402 may include a processor 2404 configured to execute logic in the form of instruction modules contained in a memory 2406, similar to processor 104 and memory 106 as shown in and described with reference to Figure 1. As with processor 104 and memory 106, processor 2404 and memory 2406 may be single devices, or may comprise multiple processors and memories having distributed functionality.

[000176] Memory 2406 may include schema for storing records 2408 therein. Records 2408 may be similar to records 108 as shown in and described with reference to Figure 1. For example, records 2408 may include user data items for multiple different users and may be defined by and stored using correlated tree structures 2410 similar to tree structures 110 as shown in and described with reference to Figure 1. In particular, tree structures 2410 may have schema definitions such that stored records 2408 are unrestricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns.

[000177] Memory 2406 may also include schema for storing privacy labels 2412 therein. Privacy labels 2412 may be similar to privacy labels 112 as shown in and described with reference to Figure 1. For example, privacy labels 2412 may distinguish among multiple different users and/or may define exclusive groupings of records 2408 for the multiple different users. Each privacy label 2412 may include a unique identification mapping to a user. In some examples, privacy label 2412 may include a number of tokens (e.g., unique randomly generated cryptographically entropic values having a predetermined number of bytes, etc.) that are uniquely provisioned to the user. Privacy labels 2412 may be correlated with each of records 2408 using privacy label tree structures included in tree structures 2410. The privacy label tree structures defining privacy labels 2412 may have schema definitions similar to other tree structures 2410. That is, privacy labels 2412 may be structured and stored without being restricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns. In some examples, records 2408 and privacy labels 2412 may be stored and maintained in a remote and/or separate multiple user data storage and separation system.

[000178] Memory 2406 may also include a workflow generation module 2414. Workflow generation module 2414 may be used to generate workflows 2416 stored in memory 2406. Workflow generation module 2414 may cause processing electronics 2402 to carry out the process 2500 set forth by the flow diagram of Figure 25. As indicated by a step 2502, workflow generation module 2414 may receive, from a resource 2420, a list of functions 2422 executable by resource 2420. For example, resource 2420 may be a case management system that allows users to enter new tickets. Users may enter new tickets by, for example, executing respective functions 2422 to enter a “Subject” for the new ticket, to enter a “Priority” for the ticket (e.g., “Low”, “Medium”, “High”, etc.), and to “Save” the new ticket. The process for entering new tickets may be a repeatable sequence of steps corresponding to the Subject, Priority, and Save functions executable by resource 2420. Resource 2420 may provide a list and/or descriptions of the Subject, Priority, and Save functions, in addition to other functions 2422, to workflow generation module 2414 to be stored locally.

[000179] At a step 2504, workflow generation module 2414 may generate a workflow 2416 including a set of functions 2422. In some examples, workflow generation module 2414 may generate workflow 2416 in response to input from a user 2440. For example, workflow generation module 2414 may provide a user interface that allows user 2440 to provide workflow data 2442 defining particular aspects of workflow 2416. In response to receiving workflow data 2442 from user 2440, workflow generation module 2414 may generate a workflow 2416 that may be stored and/or executed by apparatus 2400 and/or resource 2420.

[000180] Workflow data 2442 may include, for example, data that may be used by workflow generation module 2414 to generate a name for workflow 2416. For example, a workflow 2416 including the Subject, Priority, and Save functions executable by resource 2420 in order to create a new ticket may be named “New Ticket Workflow”. Workflow data 2442 may also include, for example, data that may be used by workflow generation module 2414 to generate a task definition step in workflow 2416 for a function 2422. The task definition step may include, for example, instructions defining a task that includes executing a particular function 2422. For example, a repeatable sequence of steps corresponding to the Subject, Priority, and Save functions executable by resource 2420 in order to create a new ticket may include three task definition steps. A first task definition step may correspond to the Subject function and may include instructions for a user execute the Subject function along with any specific instructions on how the Subject function should be executed, such as entering text describing the subject of the new ticket. The task definition step for the Subject function may also include any particular formatting or information required, a limitation on characters, etc. A second task definition step may correspond to the Priority function and may include instructions for a user to execute the Priority function along with any specific instructions on how the Subject function should be executed, such as entering “Low”, “Medium”, “High”, etc. A third task definition step may correspond to the Save function and may include instructions for a user execute the Subject function along with any specific instructions on how the Save function should be executed.

[000181] Workflow data 2442 may also include, for example, data that may be used by workflow generation module 2414 to generate a verification step in workflow 2416 for a function 2422. The verification step may define, for example, feedback and/or other evidence from resource 2420 indicating whether a particular function 2422 corresponding to a task definition step was executed. For example, a workflow 2416 including a sequence of three task definition steps corresponding to the Subject, Priority, and Save functions may include three verification steps. Each of the verification steps may correspond to a respective one of the task definition steps corresponding to the Subject, Priority, and Save functions and may define, for example, a particular audit log entry or other acceptable evidence required by apparatus 2400 in order to verify that the corresponding function 2422 has been executed by resource 2420.

[000182] Workflow data 2442 may also include, for example, data that may be used by workflow generation module 2414 to generate a response step in workflow 2416 for a function 2422. The response step may define, for example an action to be taken if function 2422 has been executed, and another action to be taken if function 2422 has not been executed. For example, a workflow 2416 including a sequence of three task definition steps corresponding to the Subject, Priority, and Save functions may include three response steps. Each of the response steps may correspond to a respective one of the task definition steps corresponding to the Subject, Priority, and Save functions. Each of the response steps may define an action to be taken by apparatus 2400 if apparatus 2400 verifies that the corresponding function 2422 has been executed by resource 2420. For example, response steps for the Subject and Priority functions may indicate an advance to the next task definition step, while a response step for the Save function may indicate the end of workflow 2416. Each of the response steps may also define an action to be taken by apparatus 2400 if apparatus 2400 does not verify that the corresponding function 2422 has been executed by resource 2420. For example, response steps corresponding to each of the Subject, Priority, and Save functions may indicate that an alert should be sent (e.g., escalating the unfinished new ticket entry to a manager, etc.).

[000183] Workflow data 2442 may also include, for example, data that may be used by workflow generation module 2414 to generate constraints governing workflow 2416. For example, workflow data 2442 may include data that may be used by workflow generation module 2414 to define a time constraint indicating an amount of time permitted to elapse between execution of functions 2422 corresponding to task definition steps in workflow 2416. For example, a workflow 2416 including a sequence of three task definition steps corresponding to the Subject, Priority, and Save functions may include a time constraint indicating a maximum amount of time that may elapse between initiating workflow 2416 and executing the Save function (e.g., a service agreement may require response times of less than a certain number of hours for entering new tickets). Similarly, workflow 2416 may also include time constraints between intermediate steps (e.g., an entity may wish to monitor the efficiency of analysts entering new tickets). Other constraints are contemplated as well. For example, workflow data 2442 may define a particular escalation path for incomplete workflows.

[000184] In some examples, the response step corresponding to a particular function 2422 may be governed by constraints. For example, a workflow 2416 including a sequence of three task definition steps corresponding to the Subject, Priority, and Save functions may include three response steps. Each of the response steps may define respective actions to be taken by apparatus 2400 if apparatus 2400 does and does not verify that the corresponding function 2422 has been executed by resource 2420. Workflow 2416 may also include a time constraint indicating a maximum amount of time that may elapse between initiating workflow 2416 and executing the Save function. Accordingly, if apparatus 2400 verifies that the Save function has been executed within the maximum amount of time set forth by the time constraint, then apparatus 2400 may take the action specified by the response step if apparatus 2400 verifies that the corresponding function 2422 has been executed by resource 2420 (e.g., log the elapsed time and advance to the next step). If apparatus 2400 does not verify that the Save function has been executed within the maximum amount of time set forth by the time constraint, then apparatus 2400 may take the action specified by the response step if apparatus 2400 does not verify that the corresponding function 2422 has been executed by resource 2420 (e.g., escalating the unfinished new ticket entry to a manager, etc.). In some examples, the time period set forth by the constraint may be adjusted to provide sufficient time for escalation to resolve the issue (e.g., a safety margin may be implemented to prevent missing a response deadline in a service agreement, etc.). Other constraint-dependent response steps are contemplated as well.

[000185] In some examples, time constraints and response steps may be used to define performance metrics for workflow 2416. Response steps may be defined, for example, to instruct apparatus 2400 to log elapsed times between the execution of two different functions 2422. For example, if a time constraint defining the maximum time between execution of two different functions 2422 is exceeded, the response step may indicate that, for example, the actual elapsed time (assuming the function is eventually executed) should be logged and tracked over time (e.g., for subsequent new tickets entered by the same user). In some examples, the number of instances where the maximum time permitted between the execution of two functions 2422 is exceeded may be tracked over time. Other performance metrics, such as actual maximum, minimum, and average elapsed times between functions, percentage of total workflow time per function, etc. may be defined as well.

[000186] Once workflow generation module 2414 generates a workflow 2416, it may be saved in memory 2406. In some examples, version control may be used to manage updates to workflow 2416. At a step 2506, workflow generation module 2414 may correlate workflow 2416 with a privacy label 2412. In some examples, workflow generation module 2424 may correlate workflow 2416 with a role that has been correlated to a particular privacy label 2412 in response to user input. Accordingly, users that are provisioned that particular privacy label (e.g., reference tokens mapped to the privacy label) may be correlated with a set of permissions corresponding to the role, and may also be correlated with a workflow 2416 associated with the role. In this way, users accessing apparatus 2400 may only access those workflows assigned to the particular role they have been provisioned via a particular privacy label 2412. For example, user 2440 may provide correlation data 2444 to workflow generation module 2414 indicating that the "New Ticket Workflow" should be assigned to users having the role "Analyst". Accordingly, workflow generation module 2414 may correlate "New Ticket Workflow" with the role "Analyst". The Analyst role may have been previously correlated with a particular privacy label 2412 that has been provisioned to a user who is an analyst having need to access resource 2420 to enter new tickets. Upon authentication and authorization of a user who has been provisioned privacy label 2412, resource 2420 may receive a mapping of the Analyst role and its corresponding set of permissions to privacy label 2412. Accordingly, when the user accesses resource 2420, resource 2420 may identify workflow 2416 as being correlated with the Analyst role mapped to privacy label 2412.

[000187] At a step 2508, workflow 2416 may be provided to resource 2420 for local storage. In some examples, workflow generation module 2414 may provide a mapping of workflow 2416 to a role to resource 2420 so that workflow 2416 may be retrieved when an appropriately provisioned user accesses resource 2420. In some examples, workflow generation module 2414 may provide resource 2420 with a resource identifier including an address where feedback from resource 2420 may be sent for purposes of executing verification steps in workflow 2416.

[000188] Figure 26 is a schematic illustration of another example workflow management apparatus 2600 for use with a multiple user data storage and separation apparatus, such as apparatus 100 shown in Figure 1. As will be described hereafter, in general, apparatus 2600 may include resource provisioning functionality. In particular, apparatus 2600 allow users to execute a workflow 2416. Apparatus 2600 is similar to apparatus 2400 shown in and described with reference to Figure 24. Those components of apparatus 2600 that correspond to apparatus 2400 are numbered similarly. Apparatus 2600 may include a workflow processing module 2615. Modules 2414 and 2615 may cooperate to cause processing electronics 2402 to carry out the process 2700 set forth by the flow diagram of Figure 27.

[000189] As indicated by a step 2702, workflow generation module 2414 may provide a workflow 2416 to a resource 2420 (e.g., as described with respect to Figure 25 and step 2508). The workflow 2416 may include task definition steps 2617 corresponding to a set of functions 2422 executable by resource 2420 (e.g., task definition steps as described with respect to Figure 25 and step 2504). The workflow 2416 may be correlated with a privacy label 2412 (e.g., as described with respect to Figure 25 and step 2506). For example, workflow generation module 2414 may provide a workflow 2416 named “New Ticket Workflow” to a resource 2420 (e.g., a case management system). A first task definition step 2617 may correspond to the Subject function and may include instructions for a user to execute the Subject function along with any specific instructions on how the Subject function should be executed, such as entering text describing the subject of the new ticket. The task definition step 2617 for the Subject function may also include any particular formatting or information required, a limitation on characters, etc. A second task definition step 2617 may correspond to the Priority function and may include instructions for a user to execute the Priority function along with any specific instructions on how the Priority function should be executed, such as entering “Low”, “Medium”, “High”, etc. A third task definition step 2617 may correspond to the Save function and may include instructions for a user execute the Save function along with any specific instructions on how the Save function should be executed.

[000190] At a step 2704, workflow processing module 2615 may receive feedback 2646 from resource 2420 indicating whether a function 2422 in the set of functions 2422 has been executed pursuant to a user 2640 providing a task response 2644 to complete a task definition step 2617 corresponding to function 2422. In some examples, feedback 2646 may be in the form of an audit log entry generated by resource 2420 upon execution of function 2422. In some examples, task response 2644 may be provided by user 2640 in response to resource 2420 providing a task definition step 2617 user 2640 (e.g., via a graphical interface displaying instructions on how to execute function 2422). In some examples, a verification step may be executed by apparatus 2600 to determine whether function 2422 has been completed (e.g., a verification step as described with reference to Figure 25 and step 2504). For example, an audit log entry may be evaluated by workflow processing module 2615 pursuant to a verification step in order to determine if a function 2422 corresponding to a task definition step 2617 has been executed. For example, the workflow 2416 named "New Ticket Workflow" may include three verification steps. Each of the verification steps may correspond to a respective one of the task definition steps 2617 corresponding to the Subject, Priority, and Save functions and may define, for example, a particular audit log entry or other acceptable evidence required by apparatus 2400 in order to verify that the corresponding function 2422 has been executed by resource 2420. For a particular instance of workflow 2416 executed for user 2640, workflow processing module 2615 may receive feedback 2646 in the form of audit logs corresponding to execution of the Subject, Priority, and Save functions.

[000191] At a step 2706, performance metrics 2618 defined in workflow 2416 may be evaluated based on feedback 2646 (e.g., performance metrics as described with respect to Figure 25 and step 2504) received from resource 2420. In some examples, feedback 2646 from resource 2420 may indicate whether two different functions 2242 in the set of functions have been executed, and workflow processing module 2615 may evaluate a performance metric 2618 defined by workflow 2416 for the two functions 2422 based on feedback 2646. In some examples, workflow processing module 2615 may evaluate a performance metric 2618 by determining an amount of time elapsed between execution of the two different functions 2422. For example, feedback 2646 corresponding to execution of the Subject, Priority, and Save functions may be received from resource 2420 for an instance of the workflow 2416 named “New Ticket Workflow”. Feedback 2646 may be evaluated by workflow processing module 2615 pursuant to verification steps corresponding to the Subject, Priority, and Save functions. Workflow processing module 2615 may determine that all three functions have been executed. Workflow processing module 2615 may log the amount of time elapsed between execution of the Subject and Priority functions, between the Priority and Save functions, and the overall execution time for workflow 2416.

[000192] Figure 28 is a schematic illustration of an example resource 2800 implementing workflow management for use with a multiple user data storage and separation apparatus, such as apparatus 100 shown in Figure 1. Apparatus 2800 may be, for example, a case management system that allows users to enter tickets, although other types of resources are contemplated as well. As will be described hereafter, in general, resource 2800 may implement workflow management functionality. In particular, apparatus 2800 may allow users to execute a workflow 2416. Apparatus 2800 may be used with, for example, apparatus 2600 shown in and described with reference to Figure 26.

[000193] Apparatus 2800 may include processing electronics 2802. Processing electronics 2802 may be similar to processing electronics 102 as shown in and described with reference to Figure 1. For example, as shown in Figure 28, processing electronics 2802 may include a processor 2804 configured to execute logic in the form of instruction modules contained in a memory 2806, similar to processor 104 and memory 106 as shown in and described with reference to Figure 1. As with processor 104 and memory 106, processor 2804 and memory 2806 may be single devices, or may comprise multiple processors and memories having distributed functionality.

[000194] Memory 2806 may include schema for storing records 2408 therein. Records 2408 may be similar to records 108 as shown in and described with reference to Figure 1. For example, records 2408 may include user data items for multiple different users and may be defined by and stored using correlated tree structures 2410 similar to tree structures 110 as shown in and described with reference to Figure 1. In particular, tree structures 2410 may have schema definitions such that stored records 2408 are unrestricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns.

[000195] Memory 2806 may also include schema for storing privacy labels 2412 therein. Privacy labels 2412 may be similar to privacy labels 112 as shown in and described with reference to Figure 1. For example, privacy labels 2412 may distinguish among multiple different users and/or may define exclusive groupings of records 2408 for the multiple different users. Each privacy label 2412 may include a unique identification mapping to a user. In some examples, privacy label 2412 may include a number of tokens (e.g., unique randomly generated cryptographically entropic values having a predetermined number of bytes, etc.) that are uniquely provisioned to the user. Privacy labels 2412 may be correlated with each of records 2408 using privacy label tree structures included in tree structures 2410. The privacy label tree structures defining privacy labels 2412 may have schema definitions similar to other tree structures 2410. That is, privacy labels 2412 may be structured and stored without being restricted by the rigid relational table structures required between instances of stored data in relational databases, such as rows and columns. In some examples, records 2408 and privacy labels 2412 may be stored and maintained in a remote and/or separate multiple user data storage and separation system.

[000196] Memory 2806 may also include functions 2422, workflows 2416, and a workflow module 2814. Workflow module 2814 may be used to generate feedback 2646 in response to execution of functions 2422. Workflow module 2814 may cause processing electronics 2802 to carry out the process 2900 set forth by the flow diagram of Figure 29. As indicated by a step 2902, workflow module 2814 may receive a workflow 2416 from a workflow management apparatus 2600. (e.g., receive a workflow 2416 provided as described with respect to Figure 25 and step 2508). The workflow 2416 may include task definition steps 2617 corresponding to a set of functions 2422 executable by resource 2420 (e.g., task definition steps as described with respect to Figure 25 and step 2504). The workflow 2416 may be correlated with a privacy label 2412 (e.g., as described with respect to Figure 25 and step 2506). For example, workflow generation module 2414 may provide a workflow 2416 named "New Ticket Workflow" to a resource 2420 (e.g., a case management system). A first task definition step 2617 may correspond to the Subject function and may include instructions for a user execute the Subject function along with any specific instructions on how the Subject function should be executed, such as entering text describing the subject of the new ticket. The task definition step 2617 for the Subject function may also include any particular formatting or information required, a limitation on characters, etc. A second task definition step 2617 may correspond to the Priority function and may include instructions for a user to execute the Priority function along with any specific instructions on how the Subject function should be executed, such as entering "Low", "Medium", "High", etc. A third task definition step 2617 may correspond to the Save function and may include instructions for a user execute the Subject function along with any specific instructions on how the Save function should be executed.

[000197] At a step 2904, workflow module 2814 may receive a request 2821 to access resource 2800 from user 2640. In some examples, workflow module 2814 may receive user identifier data from user 2640 (e.g., including reference tokens provisioned to user 2640), identify a role correlated with user 2640, and identify workflow 2416 from a set of workflows 2416 correlated with the role (e.g., as described with respect to Figure 25 and step 2506). At a step 2906, workflow module 2814 may provide task definition steps 2617 to user 2640 in response to request 2821 (e.g., provide each new step sequentially as each previous step is completed in order to guide user 2640 through workflow 2416). For example, for the workflow 2416 named "New Ticket Workflow", user 2640 may be sequentially provided with the task definition steps 2617 for the Subject, Priority, and Save functions. At a step 2908, user 2640 may provide task responses 2644. For example, user 2640 may sequentially provide a respective task response 2644 to each of the task definition steps 2617 for the Subject, Priority, and Save functions. User 2640 may, for example, respond to the task definition step 2617 for the Subject function with a text entry, to the task definition step 2617 for the Priority function with “Low”, and to the task definition step 2617 for Save by entering a save command.

[000198] At a step 2910, workflow module 2814 may provide feedback 2646 to workflow management apparatus 2600 indicating whether functions 2422 corresponding to task definition steps 2617 have been executed pursuant to the user responses 2644. In some examples, workflow module 2814 may generate audit log entries sequentially for each executed function 2422 (e.g., steps 2906, 2908, and 2910 are executed sequentially for each function 2422 in workflow 2416). In some examples, if a function 2422 corresponding to a task definition step 2617 is not completed, then process 2900 may end without advancing to the next sequential task definition step 2617. In some embodiments, workflow 2614 may be stopped and saved before all functions 2422 are executed, and may be resumed and completed at a later time.

[000199] Another example of how the apparatus, methods, and systems disclosed herein may be variously combined to facilitate functionality heretofore untenable in multi-tenancy systems is resource brokering. In particular, the resource provisioning examples disclosed herein may separate management of a host infrastructure at the operating system level from management of each hosted resource instance at the application layer while maintaining appropriate data separation and storage. Such examples may facilitate automated near real-time deployment of resources. In some examples, the resource brokering apparatus, methods, and systems disclosed herein may utilize resource provisioning functionality as well as multiple user data storage and separation functionality to facilitate automated redeployment of resources and/or transfer of data in order to leverage, for example, cost, security and/or efficiency advantages provided by incremental improvements in infrastructure technology. Such redeployments and transfers may be transparent to system users, who may continue to access data and resources using the same point of entry provided by the service provider or a separate instance manager for the resource.

[000200] By way of example, providers of cloud-based services, using the various examples of apparatus, methods, and systems for multiple data storage and separation, authentication, authorization, and resource provisioning disclosed herein, may deploy servers running an operating system capable of hosting multiple instances of resources for multiple different users (e.g., clients or other entities) with a single database storing records for each of the users. The servers may provide centralized authorization and authentication per the examples described herein, with granular access control for each of the users. The service providers may provide infrastructure management for multiple resource instances hosted by the server for multiple different clients, while at the same time allowing separate instance management for each of the resource instances, either by the service provider or a separate instance manager. The service providers may provide near real time deployment of each instance.

[000201] The service providers may desire to leverage cost, security and/or efficiency advantages provided by improvements in infrastructure technology that may be attainable through one of the other service providers. For example, a service provider offering to manage new instances of a business resource may desire to partner with another infrastructure provider that offers lower operational costs associated with hosting the resource than the service provider itself is able to offer in order to provide clients with a cost savings. In some examples, a service provider offering instance management of a particular business resource may desire to redeploy an existing instance of the business resource to a lower cost infrastructure provider while ensuring that such redeployment may be transparent to an existing client. In other examples, a service provider may wish to transfer client data from an existing infrastructure provider to a more secure infrastructure provider. In some examples, a channel partner or reseller of infrastructure management and/or instance management services may wish to partner with one or more of the service providers in order to offer clients a flexible package of infrastructure and instance management services that may transparently leverage cost, security and/or efficiency advantages provided by improvements in infrastructure technology over time that may be attainable through the other service providers.

[000202] Figure 30 is a schematic illustration of an example resource brokering apparatus 3000 that may be used with, for example, resource provisioning apparatus 1800 or 2100 of Figures 18 and 21 or components and adaptions thereof. Apparatus 3000 may be, for example, a shared resource brokering component of a cloud computing system used to provide multiple different business resources, such as email applications and case management systems, to multiple different users. In some examples, apparatus 3000 may be integrated within another apparatus such as apparatus 1800 or 2100. In some examples, apparatus 3000 may function as a central resource brokering apparatus for multiple different cloud computing service providers, some or all of which may deploy servers containing an apparatus similar to, for example, apparatus 1800 or 2100. For example, as shown in Figure 30, apparatus 3000 may be a central resource brokering apparatus for infrastructure managers 2140a, 2140b … 2140n. While apparatus 3000 is shown in Figure 30 as a separate or stand-alone device, in some examples, apparatus 3000 may be integrated with one or distributed among several or all of infrastructure managers 2140a, 2140b … 2140n. In some examples, each of several apparatus, such as apparatus 1800 or 2100, provided by each of multiple different cloud computing service providers, may utilize a separate apparatus 3000, which may be a stand-alone component or integrated therewith. Other configurations and variations are contemplated as well. As will be described hereafter, in general, apparatus 3000 may include resource brokering functionality. In particular, apparatus 3000 may facilitate transparent leveraging of cost, security and/or efficiency advantages provided by improvements in infrastructure technology among cloud computing service providers.

[000203] An infrastructure manager 2140 may include components similar to those of provisioning apparatus 1800 and apparatus 2100. As shown in detail in Figure 30 with respect to infrastructure manager 2140b, an infrastructure manager 2140 may include schema for storing records 1808 therein (e.g., in a memory 1806 as shown in Figures 18 and 21). For example, records 1808 may include user data items for multiple different users and may be defined by and stored using correlated tree structures 1810 similar to tree structures 110 as shown in and described with reference to Figure 1. Infrastructure manager 2140 may also include schema for storing privacy labels 1812 therein. Infrastructure manager 2140 may also include a resource template 1814 and a provisioning module 1816.

[000204] Apparatus 3000 may include processing electronics 3002. Processing electronics 3002 may be similar to processing electronics 102 as shown in and described with reference to Figure 1. For example, as shown in Figure 30, processing electronics 3002 may include a processor 3004 configured to execute logic in the form of instruction modules contained in a memory 3006, similar to processor 104 and memory 106 as shown in and described with reference to Figure 1. As with processor 104 and memory 106, processor 3004 and memory 3006 may be single devices, or may comprise multiple processors and memories having distributed functionality.

[000205] Memory 3006 may include a brokering module 3014. Brokering module 3014 may cause processing electronics 3002 to carry out the process 3100 set forth by the flow diagram of Figure 31. As indicated by a step 3102, brokering module 3014 may receive cost data 3022. For example, brokering module 3014 may receive cost data 3022 from published sources of pricing information and may include data related to costs of hosting a resource for an infrastructure manager 2140. In some examples, a group of cloud computing service providers may each make cost data 3022 available to brokering module 3014 in electronic format. In some examples, cost data 3022 may be provided periodically, upon pricing changes, upon request, etc. In some examples, cost data 3022 may include a detailed inventory of components and/or services required for hosting a particular resource and costs associated therewith. For example, infrastructure managers 2140 may each make cost data 3022 available to brokering module 3014 detailing components and costs associated with hosting an email application and associated client data.

[000206] In some examples, data other than cost data 3022 may be received by brokering module 3014. For example, brokering module 3014 may receive security assessment data in addition to or in place of cost data 3022. Security assessment data may include, for example, data indicative of the type and effectiveness of security measures employed by a particular infrastructure manager 2140 in order to host a particular application. In some examples, brokering module 3014 may receive efficiency data in addition to or in place of cost data 3022 and/or security assessment data. Efficiency data may include, for example, data indicative of speed, data rates, data capacity, and/or other metrics associated with hosting a particular resource.

[000207] At a step 3104, brokering module 3014 may select one of the infrastructure managers 2140 to host the resource based on an analysis of cost data 3022. The analysis may include, for example, determining hosting cost differentials among infrastructure managers 2140 using cost data 3022. Certain of infrastructure managers 2140a, 2140b … 2140n may provide cost data 3022 associated with hosting an email application to brokering module 3014. Brokering module 3014 may use cost data 3022 to calculate cost differentials among the infrastructure managers 2140 providing such cost data. In some examples, the cost differentials may be broken down by particular components and/or services required for hosting a particular resource, such as costs associated with hosting an application, costs associated with storage of data, etc. Brokering module 3014 may perform an analysis of cost data 3022 periodically, upon pricing changes, upon request, etc.

[000208] In some examples, brokering module 3014 may select one of the infrastructure managers 2140 to host the resource based on which infrastructure manager 2140 offers the lowest costs associated with hosting the resource. For example, brokering module 3014 may analyze cost data 3022 to determine cost differentials associated with hosting an email application for certain of infrastructure managers 2140a, 2140b … 2140n. Based on the analysis, brokering module may select infrastructure manager 2140b as having, for example, the lowest costs associated with hosting a particular email application. Brokering module 3014 may make a selection based on an analysis of cost data 3022 periodically, upon pricing changes, upon request, etc.

[000209] In some examples, brokering module 3014 may select one of the infrastructure managers 2140 to host the resource based on the analysis of cost data 3022 and/or additional data, such as an analysis of security assessment data and/or efficiency data. For example, based on the analysis, brokering module may select infrastructure manager 2140b as having, for example, the highest level of data security associated with hosting a particular email application. In some examples, a weighted analysis may be used where multiple types of data, such as cost, security, and efficiency data are used.

[000210] At a step 3106, brokering module 3014 may send an activation request 2152 to the selected infrastructure manager 2140 requesting activation of a resource template 1814 to host the resource. For example, brokering module 3014 may send an activation request 2152 to infrastructure manager 2140b requesting activation of a resource template 1814 to host a new instance of an email application. In some examples, the selected infrastructure manager 2140 may utilize a resource provisioning process similar to, for example, process 1900 shown in Figure 19 or process 2200 shown in Figure 22 in order to activate and configure resource template 1814. In particular, infrastructure manager 2140 may include an instruction (e.g., via provisioning module 1816) to provision a privacy label 1812 to the resource template 1814 as part of a provisioning process.

[000211] In some examples, brokering module 3014 may automatically broker a transaction (e.g., at a price consistent with cost data 3022) with the selected infrastructure manager 2140 and activation request 2152 may be automatically sent upon completion of the transaction with an infrastructure manager 2140. In some examples, activation request 2152 may be triggered by or conditioned upon other events. Brokering module 3014 may broker “one time only” transactions and perform a separate analysis prior to each transaction and/or activation request 2152 (e.g., an analysis of cost data 3022 to identify the lowest costs), or may broker transaction terms good for a particular period of time or until the occurrence of a particular event (e.g., until a lower cost provider is identified by brokering module 3014). For example, upon selecting an infrastructure manager 2140, brokering module 3014 may broker a transaction with the selected infrastructure manager, and then automatically send activation requests 2152 to infrastructure manager 2140 for new instances of a particular application for a particular period of time, after which brokering module 3014 may receive new cost, security, efficiency, or other data and perform a new analysis to identify a desired infrastructure manager 2140. Similarly, process 3100 may be initiated and/or repeated under varying conditions (e.g., periodically, upon request, for each new resource instance, data transfer, etc.).

[000212] As will be appreciated, brokering module 3014 may facilitate resource brokering among, for example, cloud computing service providers desiring to leverage cost, security and/or efficiency advantages provided by improvements in infrastructure technology that may be attainable through one of the other service providers. In some examples, a service provider offering to manage new instances of a business resource may utilize brokering module 3014 identify and broker a transaction with another infrastructure provider that offers lower operational costs associated with hosting the resource than the service provider itself is able to offer in order to provide clients with a cost savings. For example, a service provider may send a request to brokering module 3014 to identify and broker a transaction with an infrastructure manager 2140 offering the lowest costs associated with hosting an email application. Brokering module 3014 may receive cost data 3022 for multiple infrastructure managers 2140 that may offer to provide such infrastructure management. Brokering module 3014 may then analyze cost data 3022 and select the infrastructure manager 2140 offering the lowest costs associated with hosting the email application. Brokering module 3014 may then send an activation request 2152 to the selected infrastructure manager requesting activation of a resource template 1814 to host the email application. The selected infrastructure manager 2140 may utilize a resource provisioning process similar to, for example, process 1900 shown in Figure 19 or process 2200 shown in Figure 22 in order to activate and configure resource template 1814.

[000213] In some examples, a service provider offering instance management of a particular business resource may desire to redeploy an existing instance of the business resource to a lower cost infrastructure provider while ensuring that such redeployment may be transparent to an existing client. For example, a service provider having an existing email application deployed using infrastructure manager 2140a may send a request to brokering module 3014 to identify and broker a transaction with a new infrastructure manager 2140 offering lower costs associated with hosting the email application. Brokering module 3014 may receive cost data 3022 for multiple infrastructure managers 2140 that may offer to provide such infrastructure management. Brokering module 3014 may then analyze cost data 3022 and select the infrastructure manager 2140b as offering the lowest costs associated with hosting the email application. Brokering module 3014 may then send an activation request 2152 to infrastructure manager 2140b requesting activation of a resource template 1814 to host the email application. The selected infrastructure manager 2140 may utilize a resource provisioning process similar to, for example, process 1900 shown in Figure 19 or process 2200 shown in Figure 22 in order to activate and configure resource template 1814. Brokering module 3014 may also send a deactivation request to infrastructure manager 2140a requesting deactivation of the original instance of email application. Brokering module 3014 may further initiate transfer of a set of records associated with the email application from a memory location associated with infrastructure manager 2140a to a memory location associated with infrastructure manager 2140b.

[000214] Figure 32 is a schematic illustration of another example resource brokering apparatus 3200 for use with, for example, resource provisioning apparatus 2100 shown in Figure 21. As will be described hereafter, in general, apparatus 3200 may include resource brokering functionality. In particular, apparatus 3200 may facilitate transparent leveraging of cost, security and/or efficiency advantages provided by improvements in infrastructure technology among cloud computing service providers.

[000215] Apparatus 3200 may include components similar to resource brokering apparatus 3000 shown in and described with reference to Figure 30, as well as resource provisioning apparatus 2100 shown in Figure 21. Those components of apparatus 3200 that correspond to apparatus 3000 and/or apparatus 2100 are numbered similarly. In particular, a brokering module 3014 may be integrated with or otherwise be in communication with each apparatus 3200 as shown in Figure 32. While a dedicated brokering module 3014 is shown as being integrated in each resource provisioning apparatus 3200 in Figure 32, it will be appreciated that, as with apparatus 3000, brokering module 3014 may be variously centralized, dedicated, stand-alone, or distributed according to numerous examples. As shown in Figure 32, an apparatus 3200 corresponds to each of infrastructure managers 2140a and 2140b. While Figure 32 illustrates two infrastructure managers 2140, it will be understood that any suitable number of apparatus 3200 may be utilized, with each apparatus 3200 corresponding to a particular number of infrastructure managers 2140 depending on the desired configuration.

[000216] Apparatus 3200 may provide resource brokering functionality in the context of a single server running an operating system 2103. The server may host, for example, a capability-based system including a single centralized database having schema for storing records 1808 for multiple different users. The server may provide centralized authorization and authentication at the application layer with granular access control per the examples described herein using an authentication module 2110 and an authorization module 2120. The server may also host multiple resource templates 1814 that may be configured as resource instances for multiple different users (e.g., clients or other entities). For example, resource templates 1814a (I1), 1814b (I2), and 1814n (IN) may each be contained in a respective compartment of a virtual machine, and may be converted into, for example, instances of a scheduling resource a case management service provided by a cloud-based service provider and respectively corresponding to each of N different clients 2130 (e.g., clients 2130a, 2130b, ...2130n). Scheduled appointments may be created, edited, updated, and reported for each client and stored as records 1808 in the centralized database. Apparatus 3200 may also provide autonomous resource provisioning functionality similar to that described above with respect to, for example, apparatus 2100 and process 2200 shown in Figures 21 and 22.

[000217] Modules 3014 and 1816 may cooperate to cause processing electronics 1802 to carry out the process 3300 set forth by the flow diagram of Figure 33. As indicated by a step 3302, brokering module 3014 may receive data 3022, which may include, for example, cost data, security assessment data, efficiency data and/or other data associated with hosting a particular resource. For example, a service provider, such as instance manager 1824, may have an existing resource 3214, such as an email application, deployed using infrastructure manager 2140a. Instance manager 1824 may send a request to brokering module 3014 of infrastructure manager 2140b to identify and broker a transaction with a new infrastructure manager 2140 offering lower costs associated with hosting the email application so that instance manager 1824 may take advantage of the cost savings. Brokering module 3014 may receive data 3022 in a manner similar to that described with respect to step 3102 and Figure 31. For purposes of the example illustrated in Figure 33, it will be assumed that infrastructure manager 2140b will primarily execute process 3300, with some steps particularly specified as being executed by infrastructure manager 2140a. However, it will be understood that, in some examples, other distributions of the brokering functionality of process 3300 are possible as well.

[000218] At a step 3304, brokering module 3014 may analyze the received data in a manner similar to that described with respect to step 3104 and Figure 31. For example, brokering module 3014 may analyze cost data 3022 received for infrastructure managers 2140a and 2140b and determine a cost differential for hosting the email application. Brokering module 3014 may determine that infrastructure manager 2140b offers the lowest costs associated with hosting the email application. At a step 3306, brokering module 3014 may select one of the infrastructure managers 2140 to host resource 3214 based on an analysis of data 3022 in a manner similar to that described with respect to step 3104 and Figure 31. For example, brokering module 3014 may select infrastructure manager 2140b because it offers the lowest cost associated with hosting the email application.

[000219] At a step 3308, brokering module 3014 may send an activation request 2152 to the selected infrastructure manager 2140 requesting activation of a resource template 1814 to host resource 3214 in a manner similar to that described with respect to step 3106 and Figure 31. For example, brokering module 3014 may send an activation request 2152 to infrastructure manager 2140b requesting activation of a resource template 1814 to host a new instance of the email application. At a step

3310, brokering module 3014 may send a deactivation request 3224 infrastructure manager 2140. Deactivation request 3224 may, for example, request that infrastructure manager 2140a deactivate the instance of resource 3214 currently deployed thereon. For example, brokering module 3014 of infrastructure manager 2140b may send deactivation request 3224 to infrastructure manager 2140a requesting deactivation of the instance of the email application currently deployed thereon. Brokering module 3014 of infrastructure manager 2140a may then send a de-provisioning request 3226 to provisioning module 1816 requesting that resource 3214 be de-provisioned (e.g., de-provision a privacy label 1812 assigned to a resource template 1814a hosting resource 3214 on infrastructure manager 2140a.

[000220] At a step 3312, the selected infrastructure manager 2140 may utilize a resource provisioning process similar to, for example, process 1900 shown in Figure 19 or process 2200 shown in Figure 22, in order to activate and configure resource template 1814. In particular, infrastructure manager 2140 may include an instruction (e.g., via provisioning module 1816) to provision a privacy label 1812 to the resource template 1814 as part of a provisioning process. For example, provisioning module 1816 of infrastructure manager 2140b may receive a configuration request 1822 from a resource template 1814 that has been activated in response to activation request 2152. Provisioning module 1816 may respond by provisioning a privacy label 1812 to resource template 1814a and providing resource template 1814a. Provisioning module 1816 may also provide resource template 1814a with an instruction to send a configuration request 1826 to instance manager 1824. Configuration request 1826 may include the privacy label 1812 and request an application layer configuration 1828 for the email application. Instance manager 1824 may respond to configuration request with application layer configuration 1828 in order to configure the email application on infrastructure manager 2140b in resource template 1814a. Even though resource 3214 has been redeployed on infrastructure manager 2140b, clients and/or other users of the email application may still access resource 3214 via instance manager 1824 at the same point of entry.

[000221] At a step 3314, brokering module 3014 may initiate transfer of a set of records 1808 associated with resource 3214 from a memory location associated with infrastructure manager 2140a to a memory location associated with infrastructure manager 2140b. For example, brokering module 3014 may initiate transfer of a set of records 1808 associated with the email application and any privacy labels 1812 correlated therewith from a memory location associated with infrastructure manager 2140a to a memory location associated with infrastructure manager 2140b. In some examples, an intermediate staging area may be used to facilitate seamless transfer. At a step 3316, the selected infrastructure manager 2140 may receive the transferred records 1808 and any privacy labels correlated therewith, and at a step 3318, the transferred records may be correlated with the privacy label 1812 provisioned to the resource template 1814 hosting the new instance of resource 3214 on instance manager 2140b. It will be appreciated that in some examples, certain steps of process 3300 may be omitted, such as where data is being transferred without redeployment of resource 3214.

[000222] Although the present disclosure has been described with reference to example embodiments, workers skilled in the art will recognize that changes may be made in form and detail without departing from the spirit and scope of the claimed subject matter. For example, although different example embodiments may have been described as including one or more features providing one or more benefits, it is contemplated that the described features may be interchanged with one another or alternatively be combined with one another in the described example embodiments or in other alternative embodiments. Because the technology of the present disclosure is relatively complex, not all changes in the technology are foreseeable. The present disclosure described with reference to the example embodiments and set forth in the following claims is manifestly intended to be as broad as possible. For example, unless specifically otherwise noted, the claims reciting a single particular element also encompass a plurality of such particular elements.

WHAT IS CLAIMED IS:

1. A resource brokering apparatus, comprising:

memory; and

instructions stored in the memory, the instructions, when executed, to cause a processor to

receive, for multiple different infrastructure managers, cost data associated with hosting a resource;

select one of the infrastructure managers to host the resource based on an analysis of the cost data; and

send an activation request to the selected infrastructure manager requesting activation of a resource template to host the resource, the infrastructure manager including an instruction to provision a privacy label to the resource template, the privacy label correlatable with privacy labels stored in a database and defined by correlated tree structures having schema unrestricted by relational table structures, the privacy labels correlatable with records including user data items associated with multiple different users, and the privacy labels distinguishing among the users.

2. The apparatus of claim 1, wherein a first instance of the resource is hosted by a first infrastructure manager, wherein the cost data includes cost data for the first infrastructure manager and a second infrastructure manager, and wherein the instructions, when executed, cause the processor to select one of the first and second infrastructure managers based on an analysis of the cost data, and, in response to the processor selecting the second infrastructure manager, to send the activation request to the second infrastructure manager, the second activation request requesting activation of the resource template to host a second instance of the resource.

3. The apparatus of claim 2, wherein the instructions, when executed, cause the processor to send a deactivation request to the first infrastructure manager requesting deactivation of the first instance of the resource, the first infrastructure manager including an instruction to de-provision the resource in response to the deactivation request.

4. The apparatus of claim 2, wherein the instructions, when executed, cause the processor to initiate transfer of a set of records associated with the resource from a first memory location associated with the first infrastructure manager to a second memory location associated with the second infrastructure manager.

5. The apparatus of claim 1, wherein the infrastructure manager includes an instruction to receive a first configuration request from the resource template and to respond to the resource template with an instruction to send a second configuration request to an instance manager, the second configuration request including the privacy label and requesting an application layer configuration for the resource.

6. A resource brokering method, comprising:

receiving cost data associated with hosting a resource from infrastructure managers;

selecting one of the infrastructure managers to host the resource based on an analysis of the cost data; and

sending an activation request to the selected infrastructure manager requesting activation of a resource template to host the resource, the infrastructure manager including an instruction to provision a privacy label to the resource template, the privacy label correlatable with privacy labels stored in a database and defined by correlated tree structures having schema unrestricted by relational table structures, the privacy labels correlatable with records including user data items associated with multiple different users, and the privacy labels distinguishing among the users.

7. The method of claim 6, further comprising determining hosting cost differentials among the infrastructure managers in order to provide the analysis of the cost data.

8. The method of claim 7, wherein selecting the one of the infrastructure managers includes selecting the infrastructure manager having the lowest costs associated with hosting the resource.

9. The method of claim 6, further comprising receiving one of security assessment data and efficiency data associated with hosting the resource from the infrastructure managers.

10. The method of claim 9, further comprising selecting the one of the infrastructure managers to host the resource based on the analysis of the cost data and an analysis of the one of the security assessment data and the efficiency data.

11. A resource brokering system, comprising:

a processor;

memory in communication with the processor; and

instructions stored in the memory for directing the processor, the instructions including

a brokering module to receive, for multiple different infrastructure managers, cost data associated with hosting a resource, to select one of the infrastructure managers to host the resource based on an analysis of the cost data, and to send an activation request to the selected infrastructure manager requesting activation of a resource template to host the resource; and

a provisioning module to provision a privacy label to the resource template, the privacy label correlatable with privacy labels stored in a database and defined by correlated tree structures having schema unrestricted by relational table structures, the privacy labels correlatable with records including user data items associated with multiple different users, and the privacy labels distinguishing among the users.

12. The system of claim 11, wherein a first instance of the resource is hosted by a first infrastructure manager, wherein the cost data includes cost data for the first infrastructure manager and a second infrastructure manager, and wherein the brokering module includes an instruction to select one of the first and second infrastructure managers based on an analysis of the cost data, and, in response to the processor selecting the second infrastructure manager, to send the activation request to the second infrastructure manager, the second activation request requesting activation of the resource template to host a second instance of the resource.

13. The system of claim 12, wherein the brokering module includes an instruction to send a deactivation request to the first infrastructure manager requesting deactivation of the first instance of the resource, the first infrastructure manager including an instruction to de-provision the resource in response to the deactivation request.

14. The system of claim 12, wherein the brokering module includes an instruction to initiate transfer of a set of records associated with the resource from a first memory location associated with the first infrastructure manager to a second memory location associated with the second infrastructure manager, and wherein the provisioning module includes an instruction to correlate the set of records with the privacy label provisioned to the resource template.

15. The system of claim 11, wherein the provisioning module includes an instruction to receive a first configuration request from the resource template and to respond to the resource template with an instruction to send a second configuration request to an instance manager, the second configuration request including the privacy label and requesting an application layer configuration for the resource.

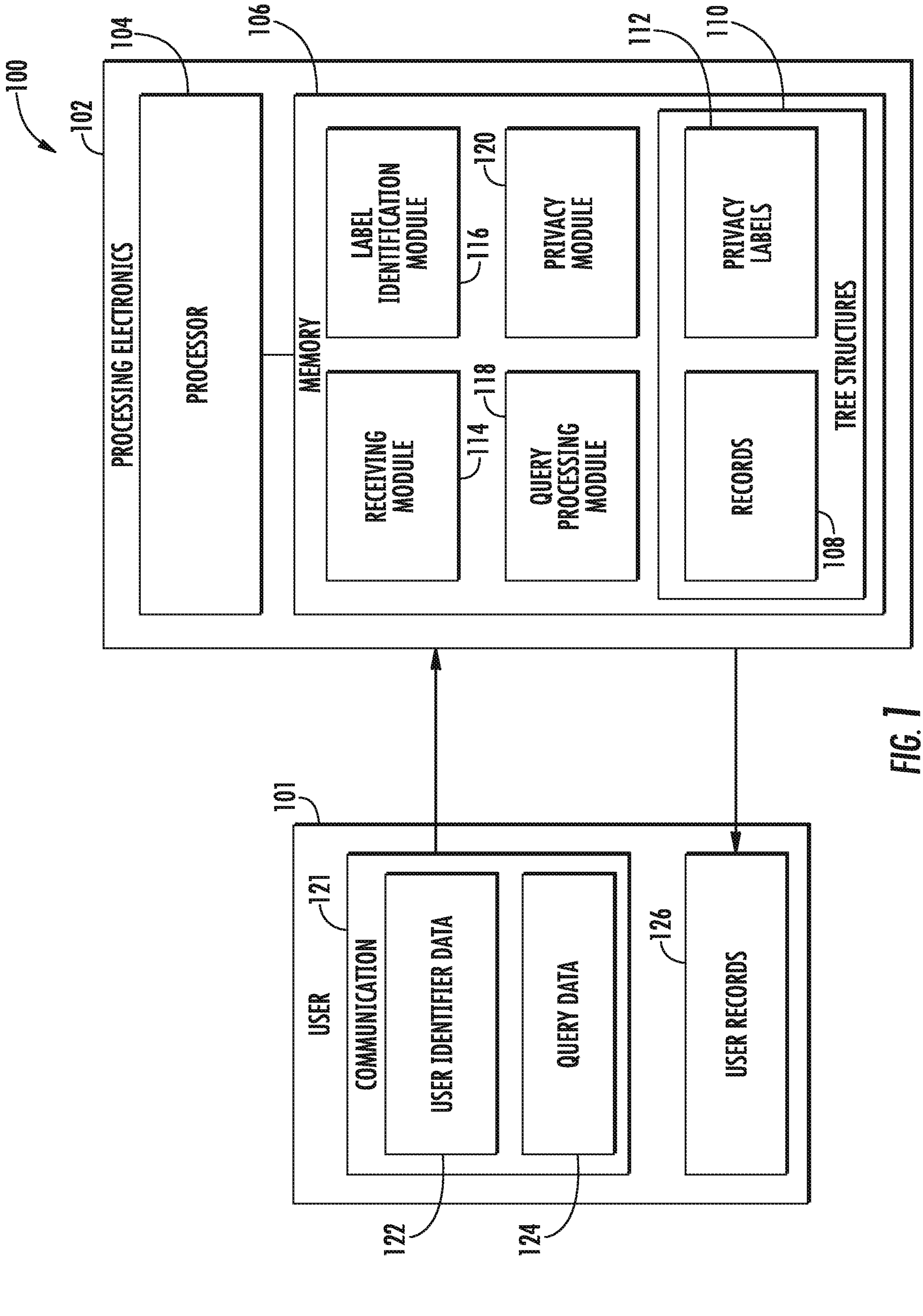
100
102
PROCESSING ELECTRONICS
104
PROCESSOR
106
MEMORY
LABEL IDENTIFICATION MODULE
116
120
PRIVACY MODULE
112
110
PRIVACY LABELS
RECEIVING MODULE
114
118
QUERY PROCESSING MODULE
RECORDS
108
TREE STRUCTURES
101
USER
121
COMMUNICATION
USER IDENTIFIER DATA
QUERY DATA
126
USER RECORDS
122
124

FIG. 1

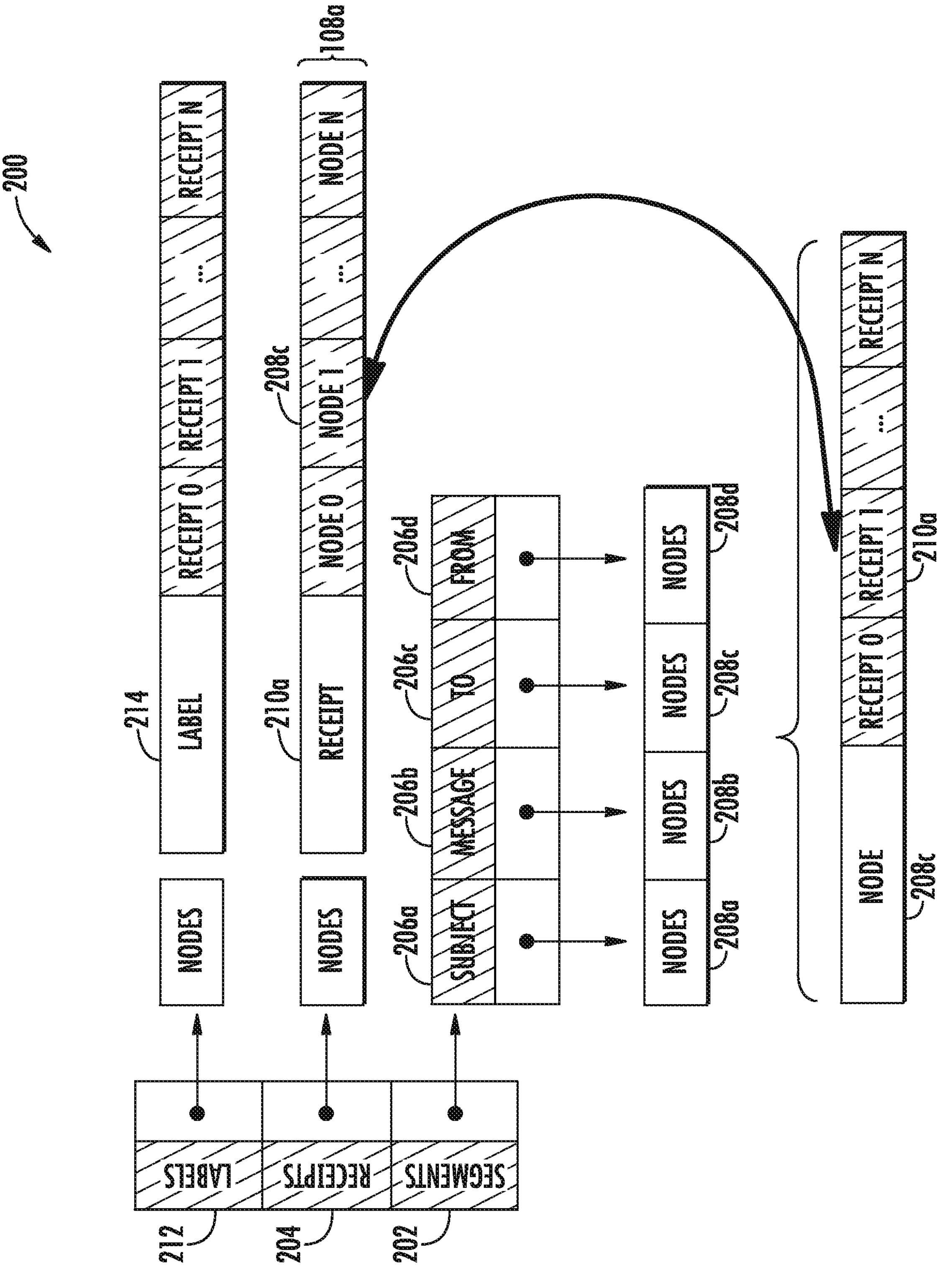
200
212
LABELS
204
RECEIPTS
202
SEGMENTS
NODES
NODES
214
LABEL
RECEIPT 0
RECEIPT 1
...
RECEIPT N
210a
RECEIPT
NODE 0
208c
NODE 1
...
NODE N
108a
206a
SUBJECT
206b
MESSAGE
206c
TO
206d
FROM
NODES
208a
NODES
208b
NODES
208c
NODES
208d
NODE
208c
RECEIPT 0
RECEIPT 1
210a
...
RECEIPT N

FIG. 2

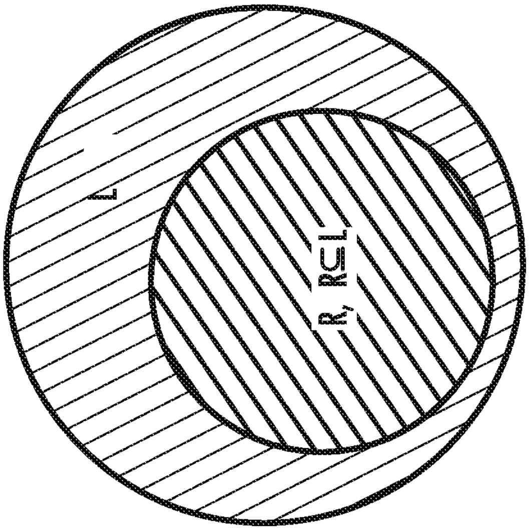

FIG. 3A

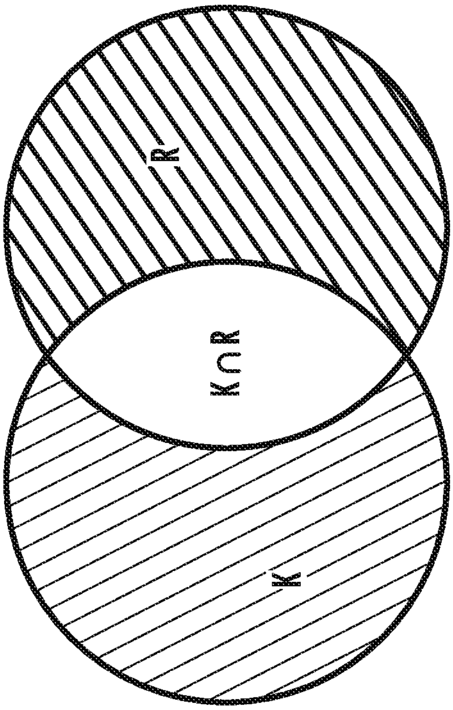

FIG. 3B

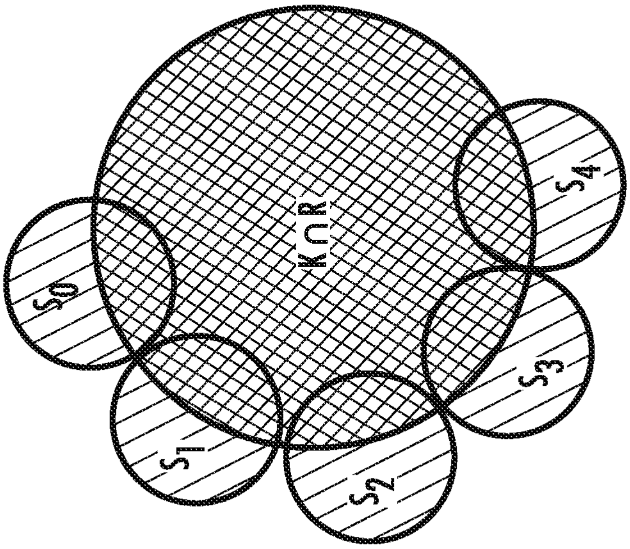

FIG. 3C

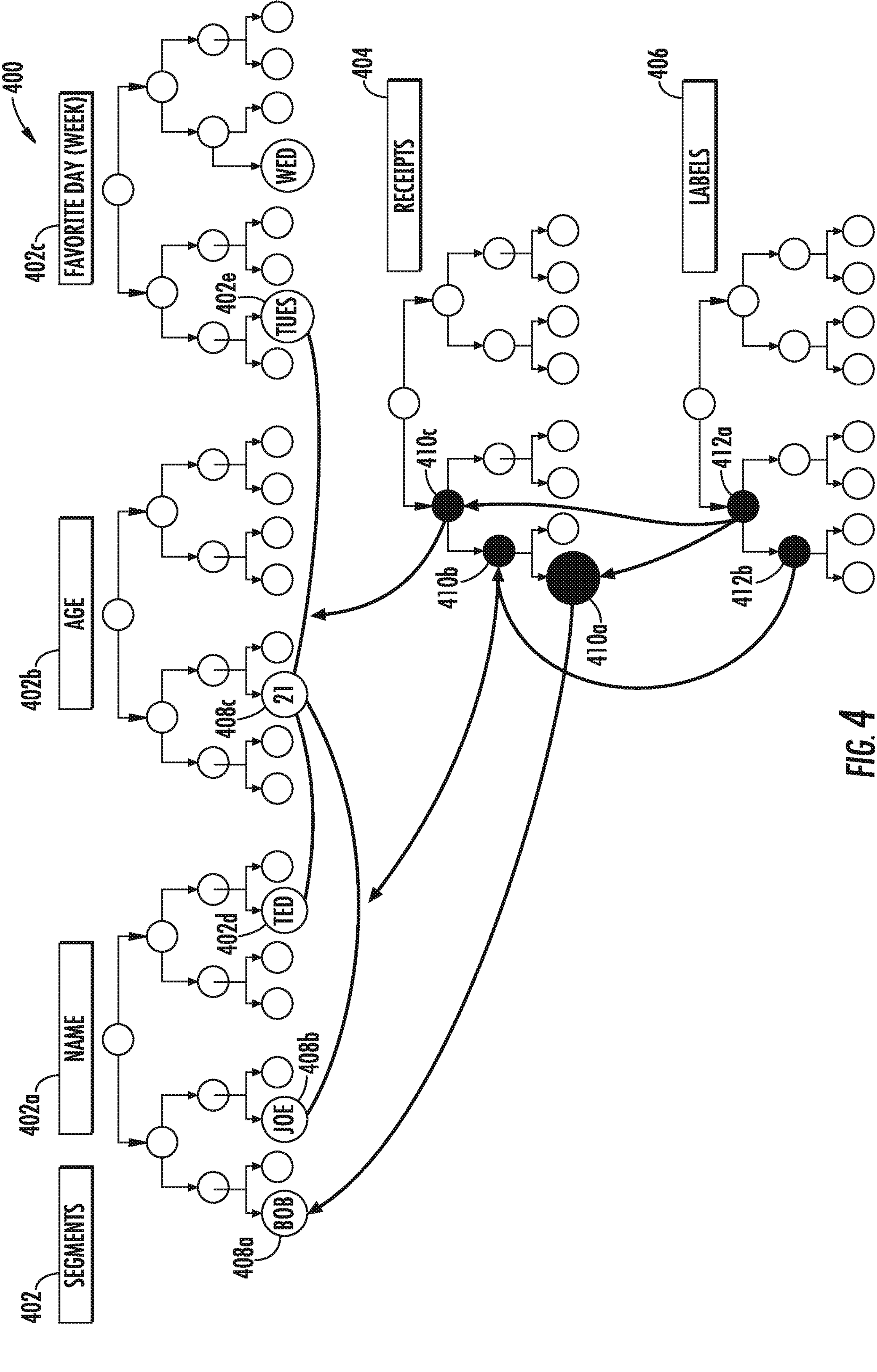
400
402
SEGMENTS
402a
NAME
402b
AGE
402c
FAVORITE DAY (WEEK)
404
RECEIPTS
406
LABELS
BOB
408a
JOE
408b
TED
402d
21
408c
TUES
402e
WED
410a
410b
410c
412a
412b

FIG. 4

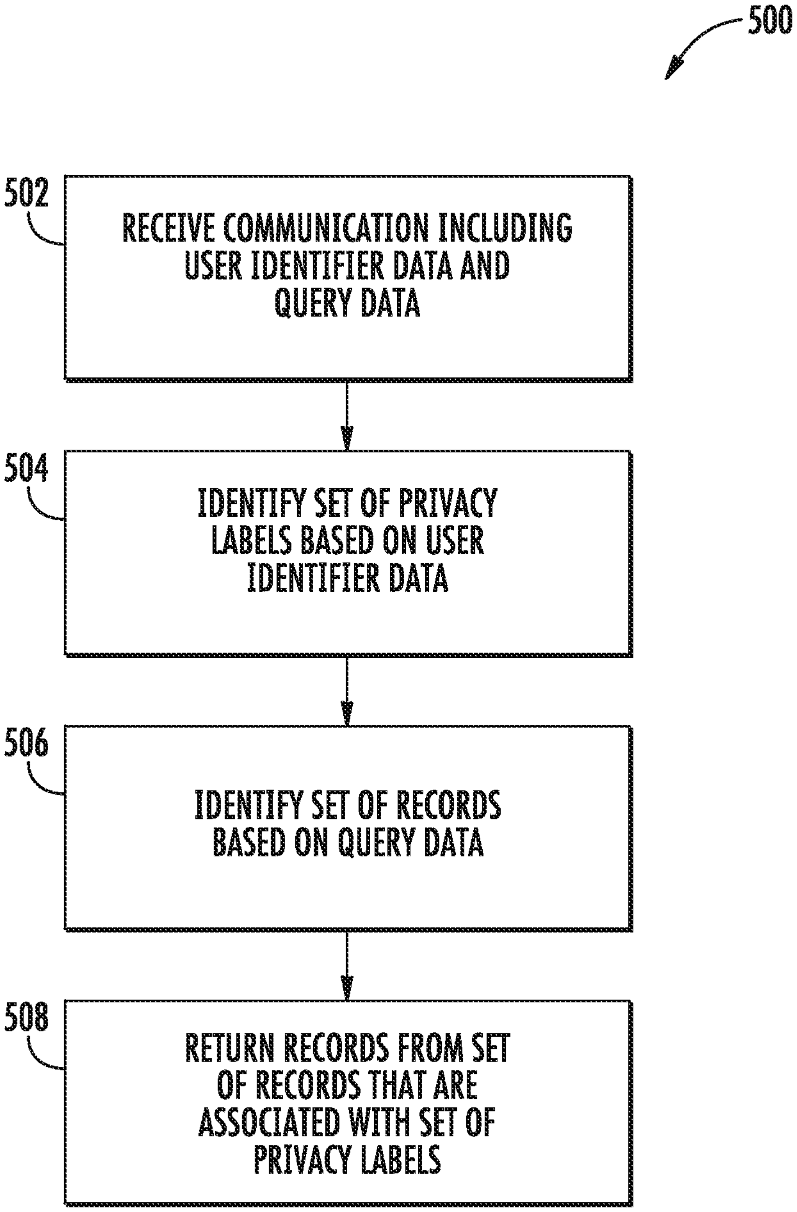
500
502
RECEIVE COMMUNICATION INCLUDING USER IDENTIFIER DATA AND QUERY DATA
504
IDENTIFY SET OF PRIVACY LABELS BASED ON USER IDENTIFIER DATA
506
IDENTIFY SET OF RECORDS BASED ON QUERY DATA
508
RETURN RECORDS FROM SET OF RECORDS THAT ARE ASSOCIATED WITH SET OF PRIVACY LABELS

FIG. 5

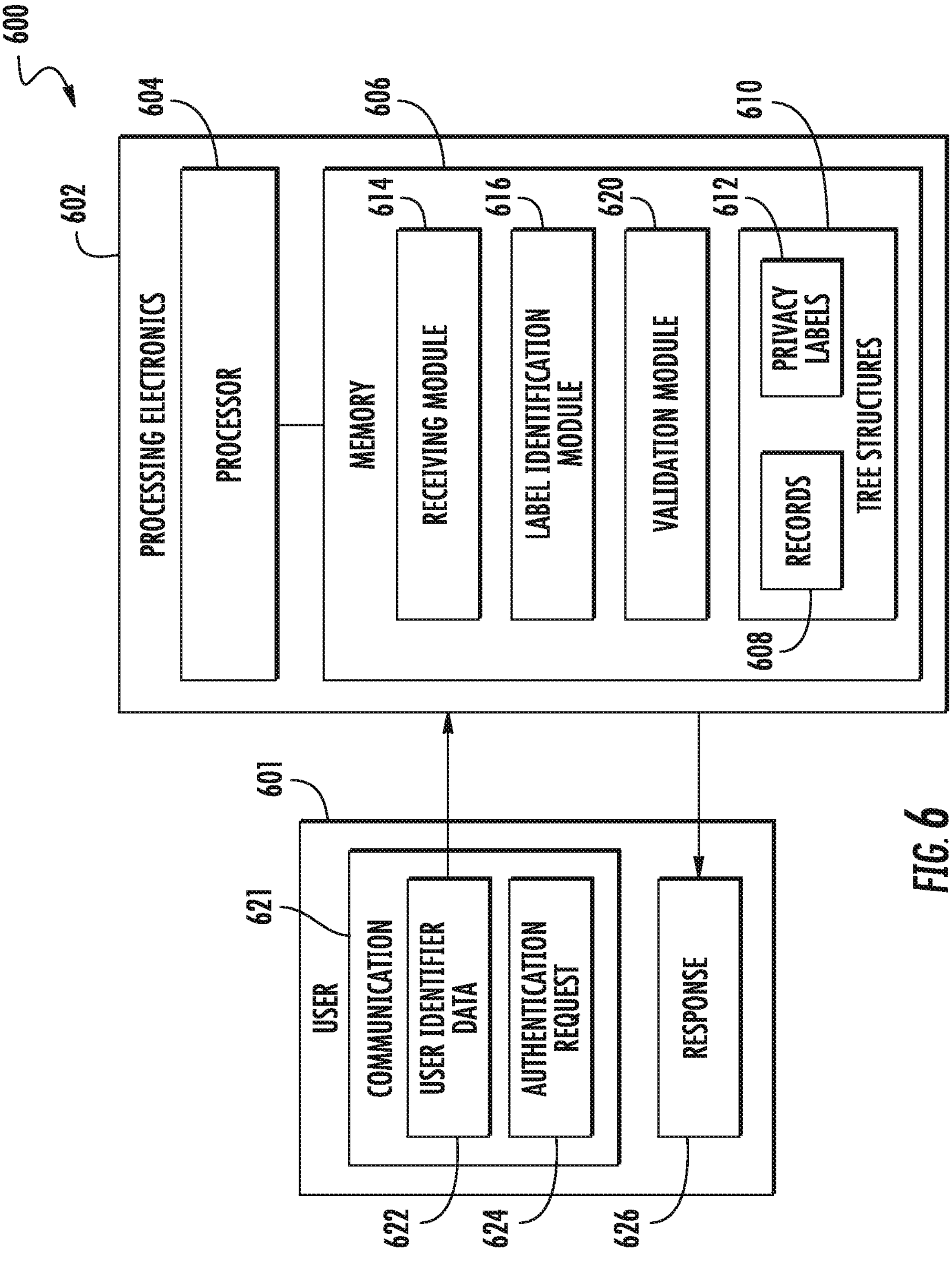
600
602
PROCESSING ELECTRONICS
604
PROCESSOR
606
MEMORY
614
RECEIVING MODULE
616
LABEL IDENTIFICATION MODULE
620
VALIDATION MODULE
610
612
PRIVACY LABELS
RECORDS
608
TREE STRUCTURES
601
USER
621
COMMUNICATION
USER IDENTIFIER DATA
622
AUTHENTICATION REQUEST
624
RESPONSE
626

FIG. 6

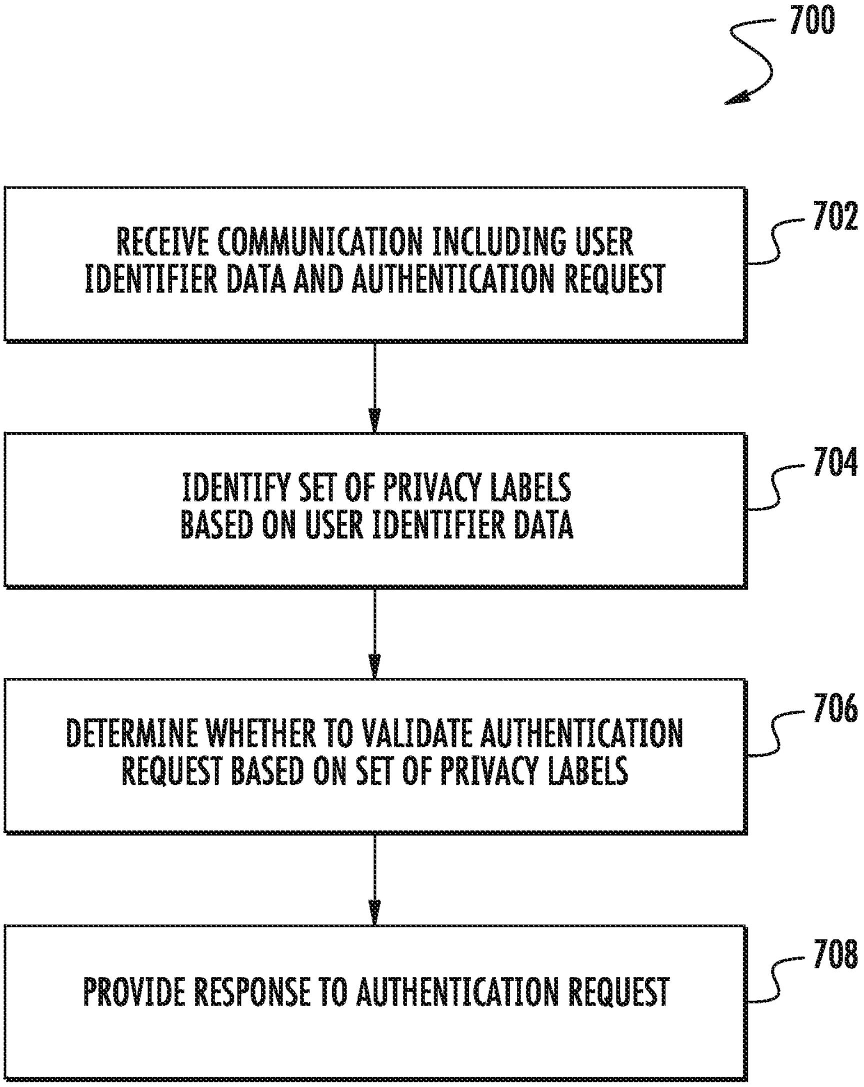
700
RECEIVE COMMUNICATION INCLUDING USER IDENTIFIER DATA AND AUTHENTICATION REQUEST
702
IDENTIFY SET OF PRIVACY LABELS BASED ON USER IDENTIFIER DATA
704
DETERMINE WHETHER TO VALIDATE AUTHENTICATION REQUEST BASED ON SET OF PRIVACY LABELS
706
PROVIDE RESPONSE TO AUTHENTICATION REQUEST
708

FIG. 7

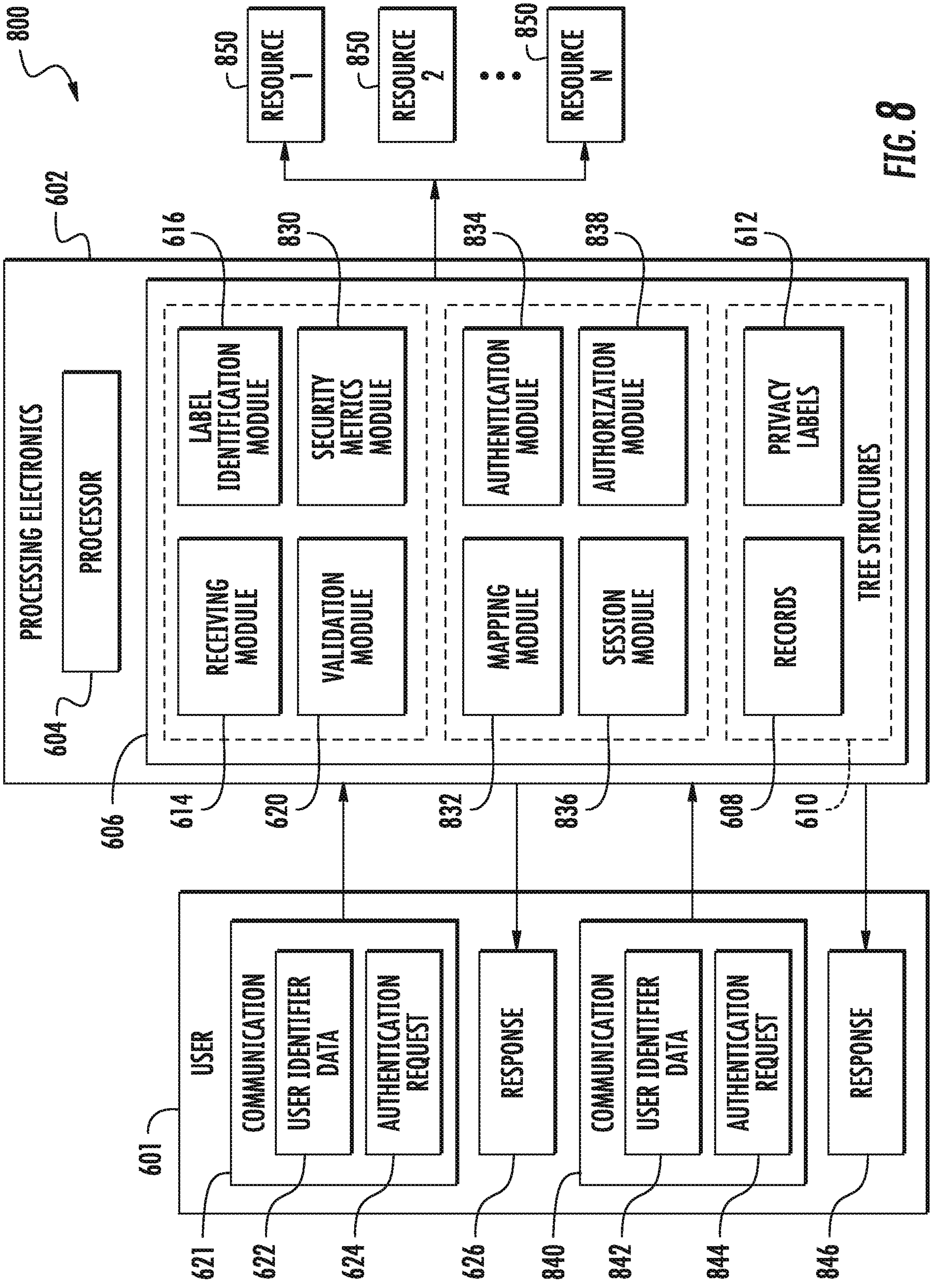
800
850
RESOURCE 1
850
RESOURCE 2
850
RESOURCE N
602
PROCESSING ELECTRONICS
PROCESSOR
604
606
616
LABEL IDENTIFICATION MODULE
RECEIVING MODULE
614
830
SECURITY METRICS MODULE
VALIDATION MODULE
620
834
AUTHENTICATION MODULE
MAPPING MODULE
832
838
AUTHORIZATION MODULE
SESSION MODULE
836
612
PRIVACY LABELS
RECORDS
608
TREE STRUCTURES
610
601
USER
COMMUNICATION
USER IDENTIFIER DATA
AUTHENTICATION REQUEST
RESPONSE
COMMUNICATION
USER IDENTIFIER DATA
AUTHENTICATION REQUEST
RESPONSE
621
622
624
626
840
842
844
846

FIG. 8

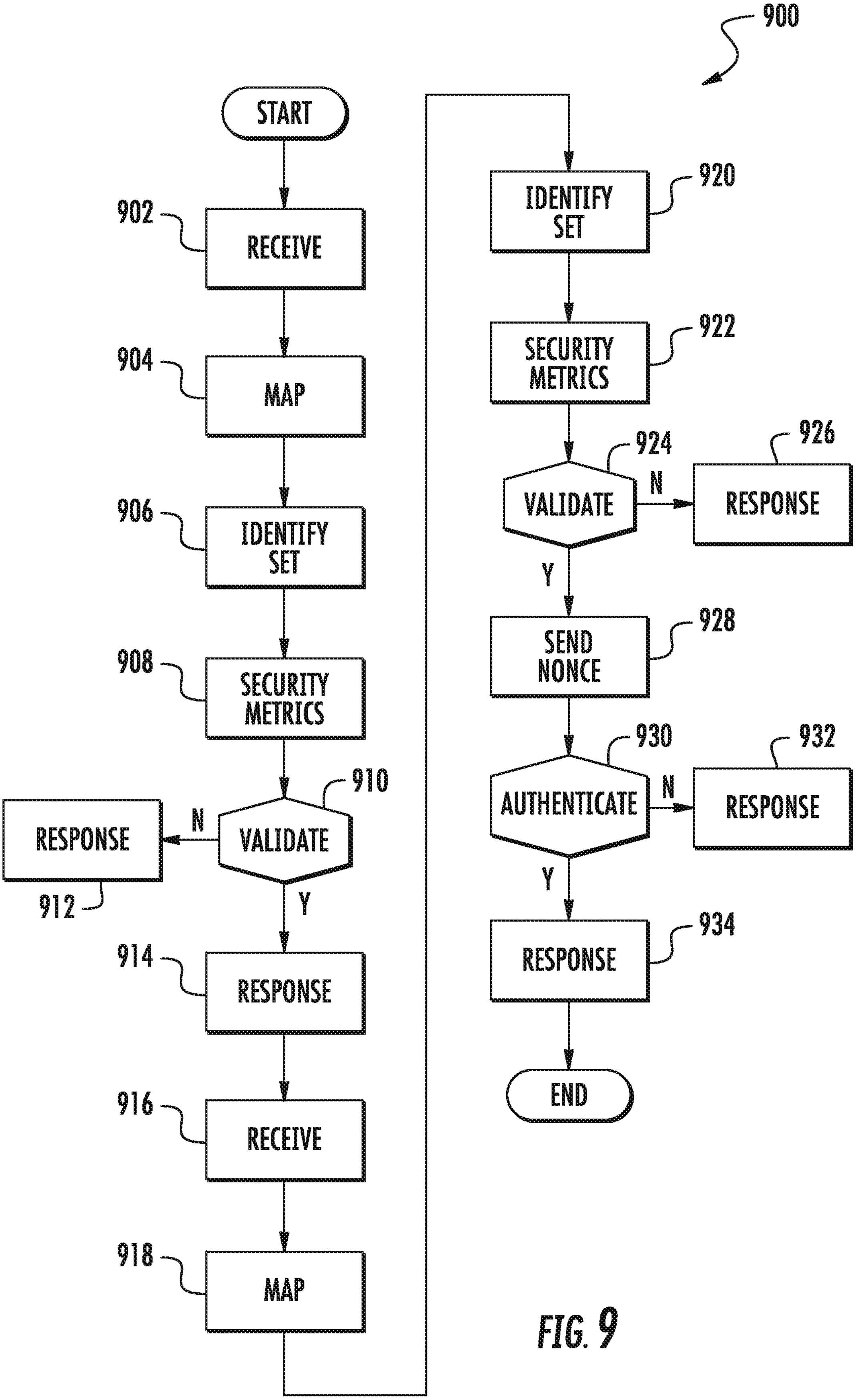
900
START
902
RECEIVE
904
MAP
906
IDENTIFY SET
908
SECURITY METRICS
910
VALIDATE
N
RESPONSE
912
Y
914
RESPONSE
916
RECEIVE
918
MAP
920
IDENTIFY SET
922
SECURITY METRICS
924
VALIDATE
N
926
RESPONSE
Y
928
SEND NONCE
930
AUTHENTICATE
N
932
RESPONSE
Y
934
RESPONSE
END

FIG. 9

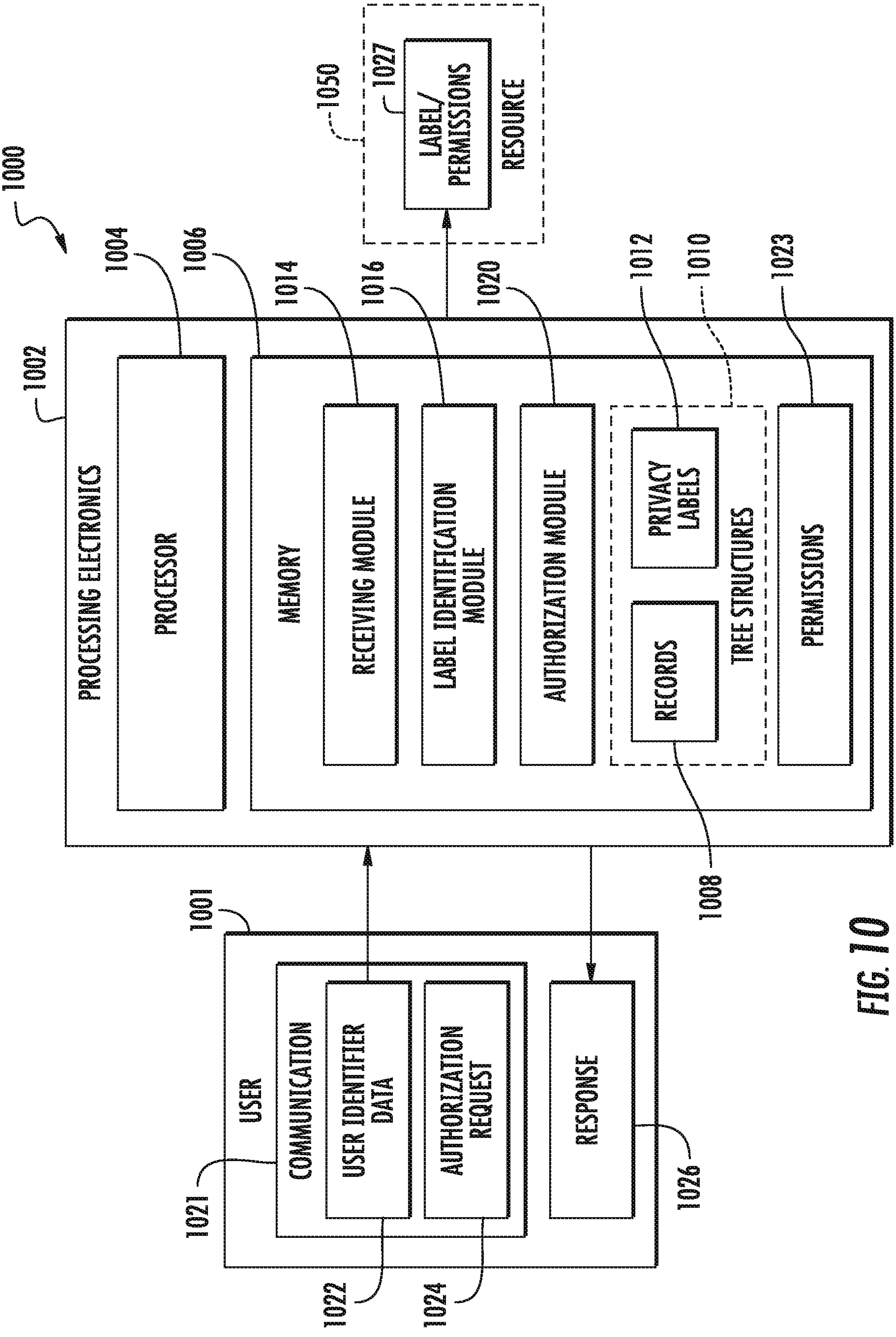
1000
1002
PROCESSING ELECTRONICS
1004
PROCESSOR
1006
MEMORY
1014
RECEIVING MODULE
1016
LABEL IDENTIFICATION MODULE
1020
AUTHORIZATION MODULE
1012
1010
PRIVACY LABELS
RECORDS
TREE STRUCTURES
1008
1023
PERMISSIONS
1050
1027
LABEL/PERMISSIONS
RESOURCE
1001
USER
1021
COMMUNICATION
USER IDENTIFIER DATA
AUTHORIZATION REQUEST
1022
1024
RESPONSE
1026

FIG. 10

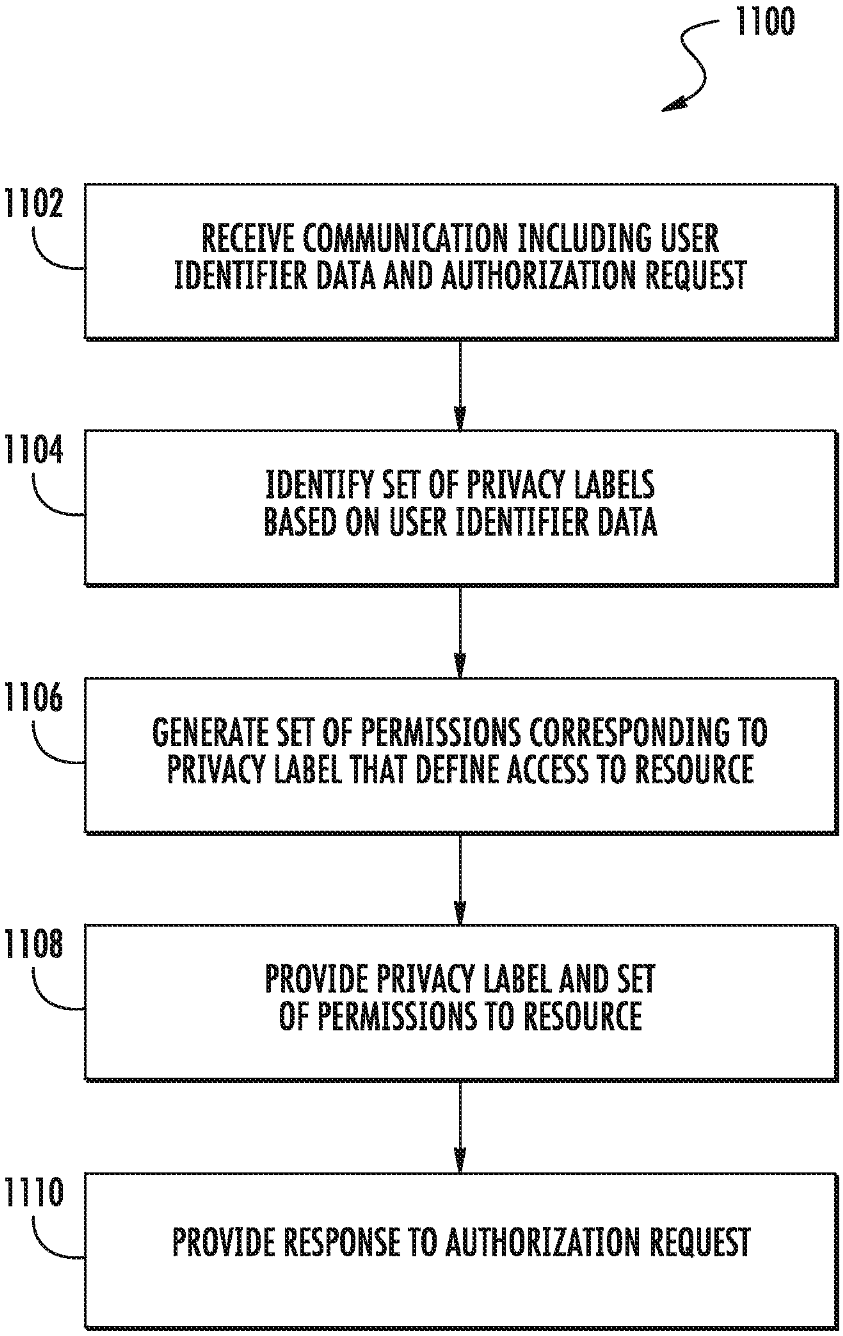
1100
1102
RECEIVE COMMUNICATION INCLUDING USER IDENTIFIER DATA AND AUTHORIZATION REQUEST
1104
IDENTIFY SET OF PRIVACY LABELS BASED ON USER IDENTIFIER DATA
1106
GENERATE SET OF PERMISSIONS CORRESPONDING TO PRIVACY LABEL THAT DEFINE ACCESS TO RESOURCE
1108
PROVIDE PRIVACY LABEL AND SET OF PERMISSIONS TO RESOURCE
1110
PROVIDE RESPONSE TO AUTHORIZATION REQUEST

FIG. 11

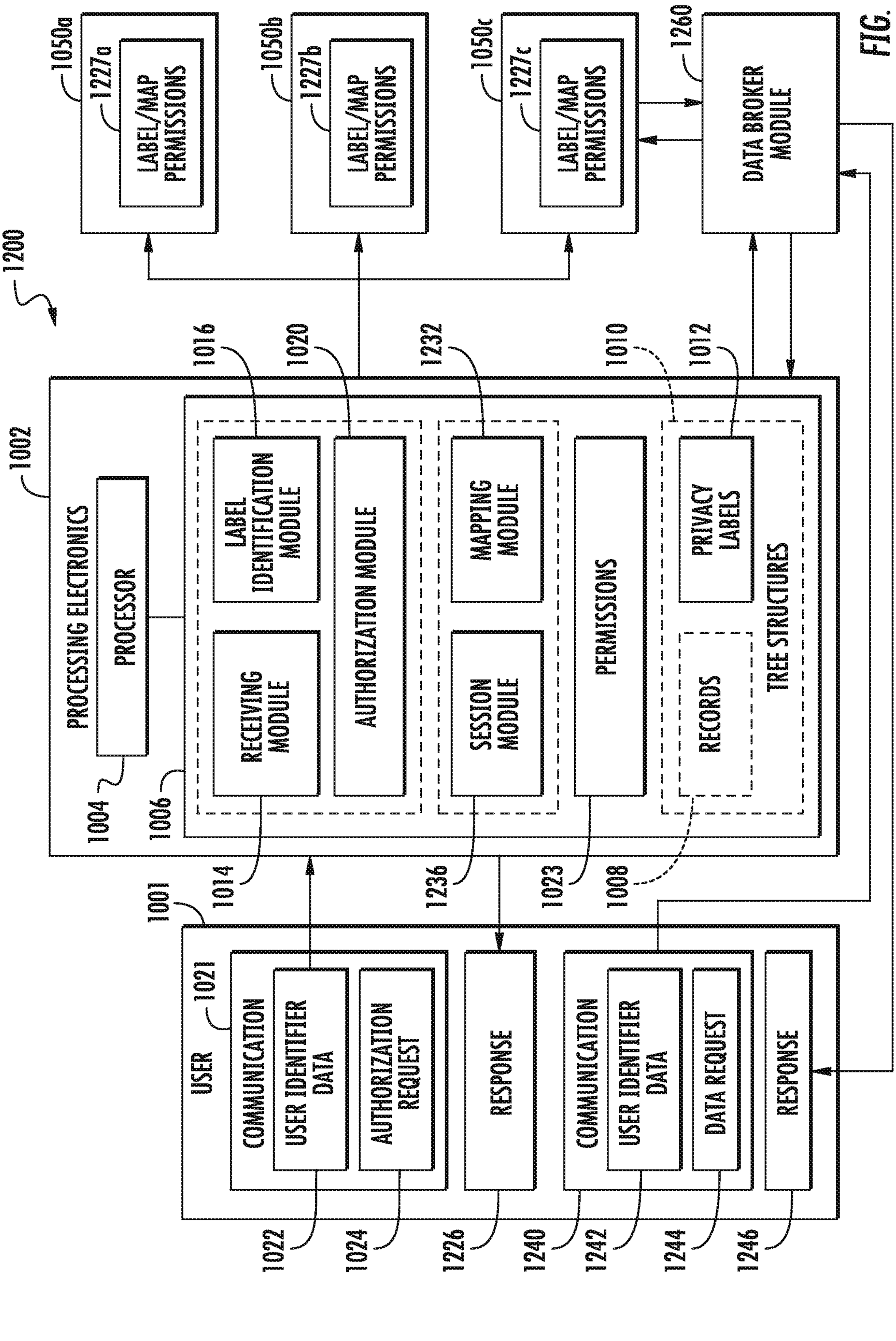
1200
1050a
1227a
LABEL/MAP PERMISSIONS
1050b
1227b
LABEL/MAP PERMISSIONS
1050c
1227c
LABEL/MAP PERMISSIONS
1260
DATA BROKER MODULE
1002
PROCESSING ELECTRONICS
PROCESSOR
1004
1006
1016
1020
1232
1010
1012
LABEL IDENTIFICATION MODULE
RECEIVING MODULE
AUTHORIZATION MODULE
MAPPING MODULE
SESSION MODULE
PERMISSIONS
PRIVACY LABELS
RECORDS
TREE STRUCTURES
1014
1236
1023
1008
1001
USER
1021
COMMUNICATION
USER IDENTIFIER DATA
AUTHORIZATION REQUEST
RESPONSE
COMMUNICATION
USER IDENTIFIER DATA
DATA REQUEST
RESPONSE
1022
1024
1226
1240
1242
1244
1246

FIG. 12

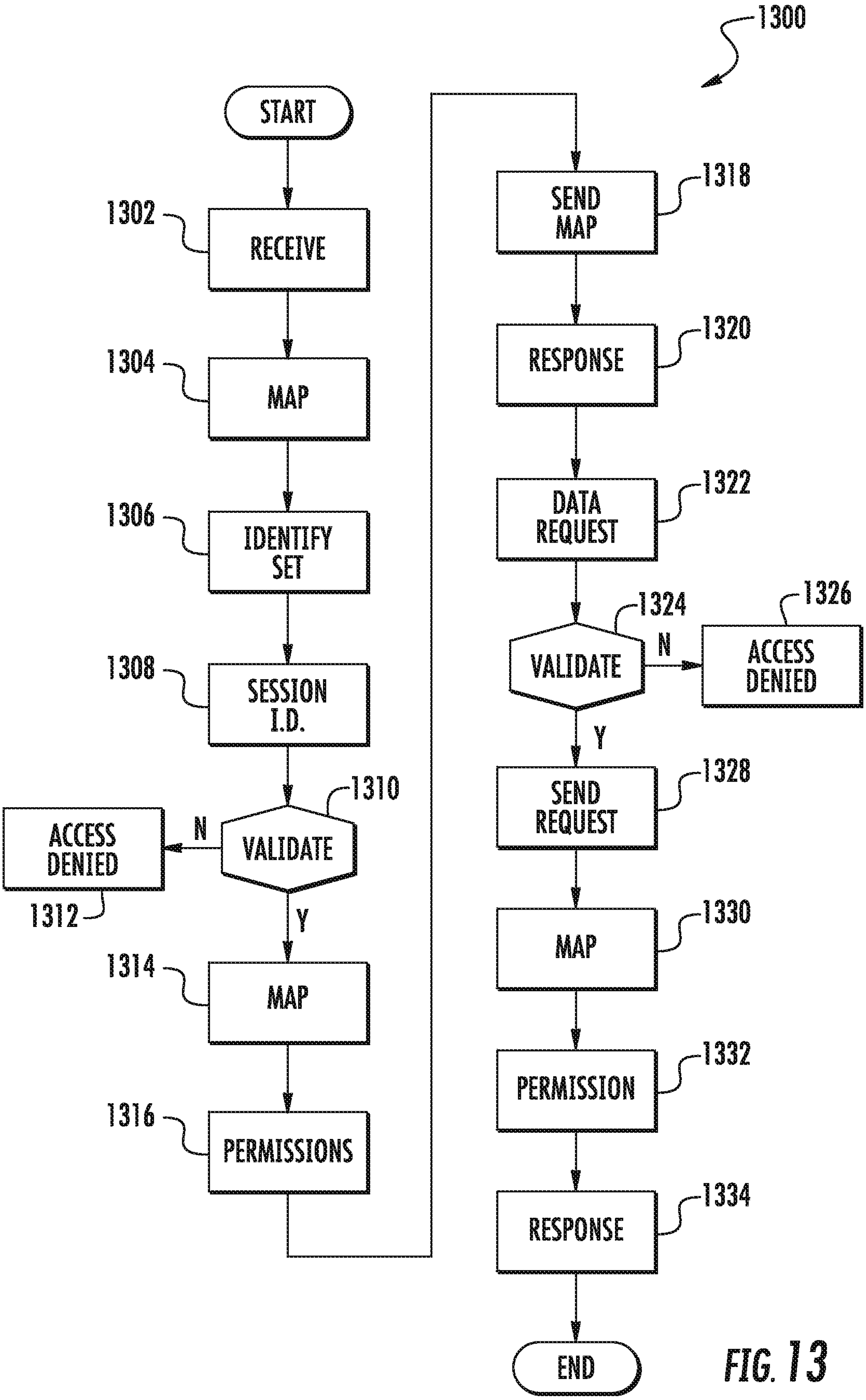
1300
START
1302
RECEIVE
1304
MAP
1306
IDENTIFY SET
1308
SESSION I.D.
1310
VALIDATE
N
ACCESS DENIED
1312
Y
1314
MAP
1316
PERMISSIONS
1318
SEND MAP
1320
RESPONSE
1322
DATA REQUEST
1324
VALIDATE
N
1326
ACCESS DENIED
Y
1328
SEND REQUEST
1330
MAP
1332
PERMISSION
1334
RESPONSE
END

FIG. 13

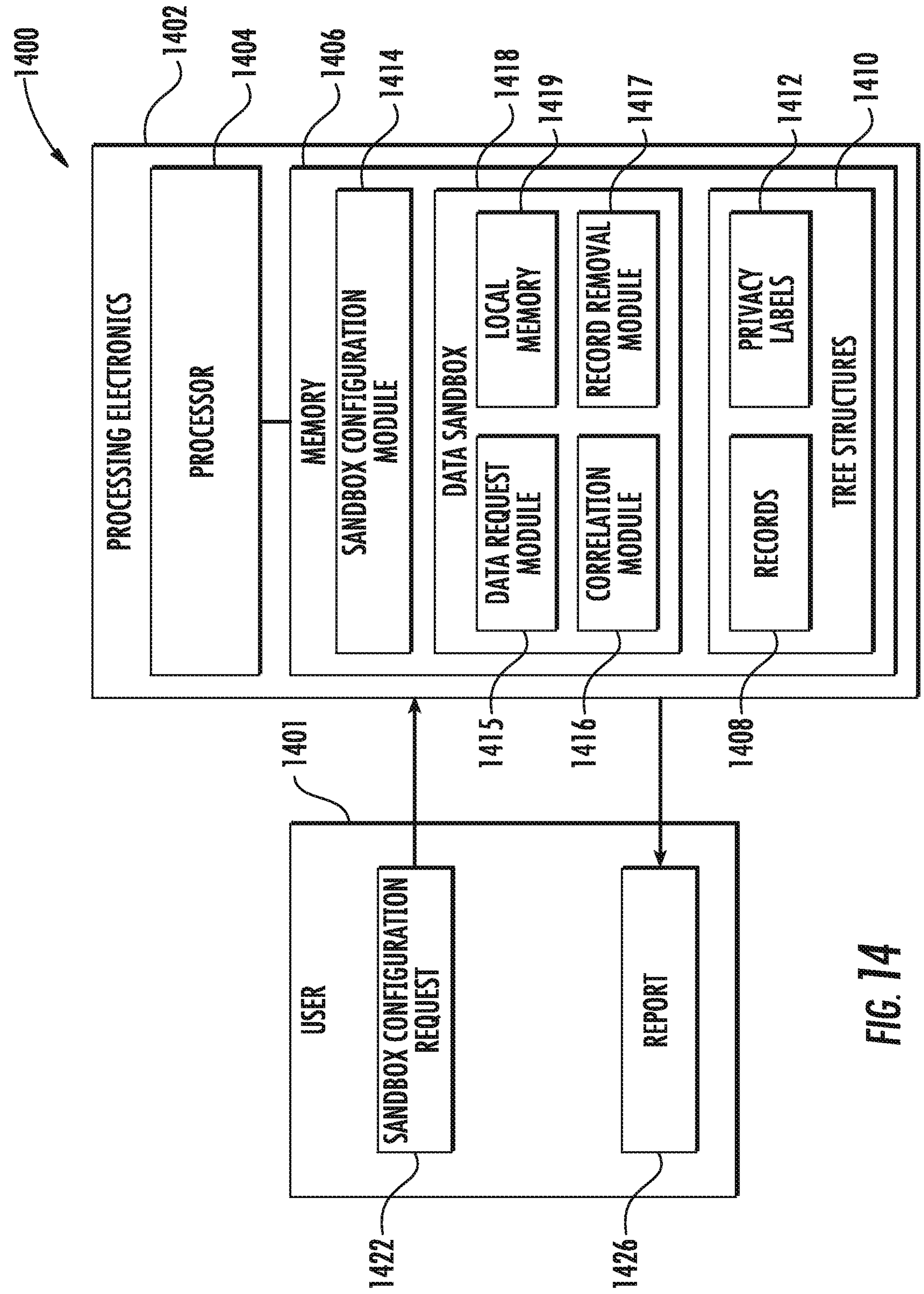
1400
1402
1404
1406
1414
1418
1419
1417
1412
1410
PROCESSING ELECTRONICS
PROCESSOR
MEMORY
SANDBOX CONFIGURATION MODULE
DATA SANDBOX
LOCAL MEMORY
RECORD REMOVAL MODULE
DATA REQUEST MODULE
CORRELATION MODULE
PRIVACY LABELS
RECORDS
TREE STRUCTURES
1415
1416
1408
1401
USER
SANDBOX CONFIGURATION REQUEST
REPORT
1422
1426

FIG. 14

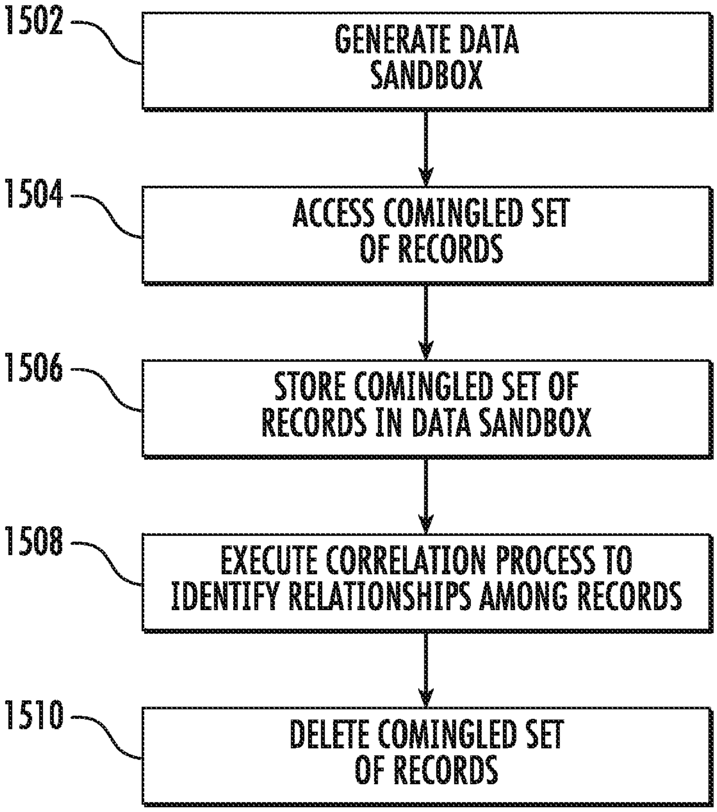
1502
GENERATE DATA SANDBOX
1504
ACCESS COMINGLED SET OF RECORDS
1506
STORE COMINGLED SET OF RECORDS IN DATA SANDBOX
1508
EXECUTE CORRELATION PROCESS TO IDENTIFY RELATIONSHIPS AMONG RECORDS
1510
DELETE COMINGLED SET OF RECORDS

FIG. 15

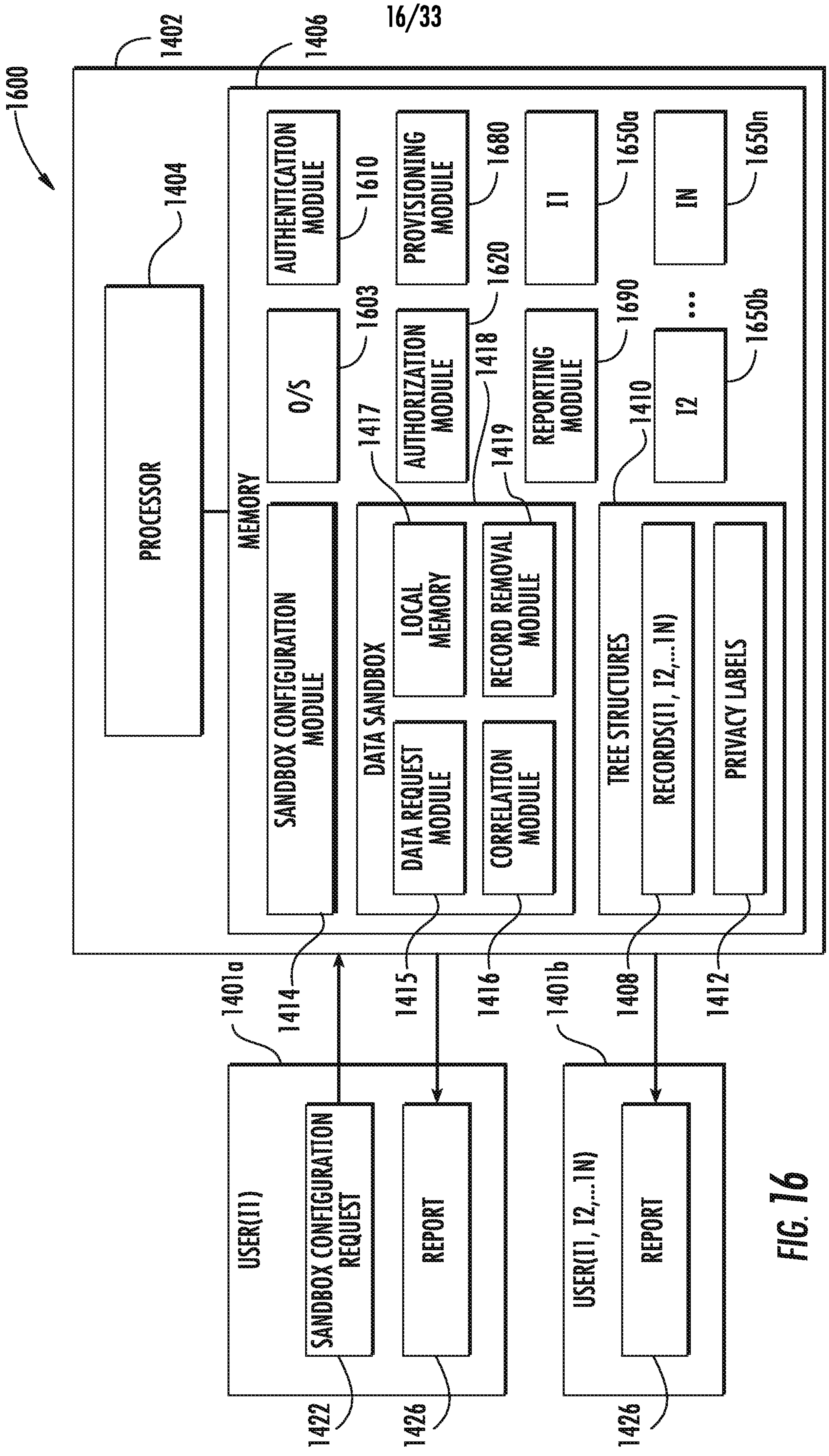
1600
1402
1404
PROCESSOR
1406
MEMORY
AUTHENTICATION MODULE
1610
PROVISIONING MODULE
1680
I1
1650a
IN
1650n
O/S
1603
AUTHORIZATION MODULE
1620
REPORTING MODULE
1690
I2
1650b
1410
1417
1418
1419
SANDBOX CONFIGURATION MODULE
DATA SANDBOX
LOCAL MEMORY
RECORD REMOVAL MODULE
DATA REQUEST MODULE
CORRELATION MODULE
TREE STRUCTURES
RECORDS(I1, I2,...IN)
PRIVACY LABELS
1414
1415
1416
1408
1412
1401a
1401b
USER(I1)
SANDBOX CONFIGURATION REQUEST
REPORT
1422
1426
USER(I1, I2,...IN)
REPORT
1426

FIG. 16

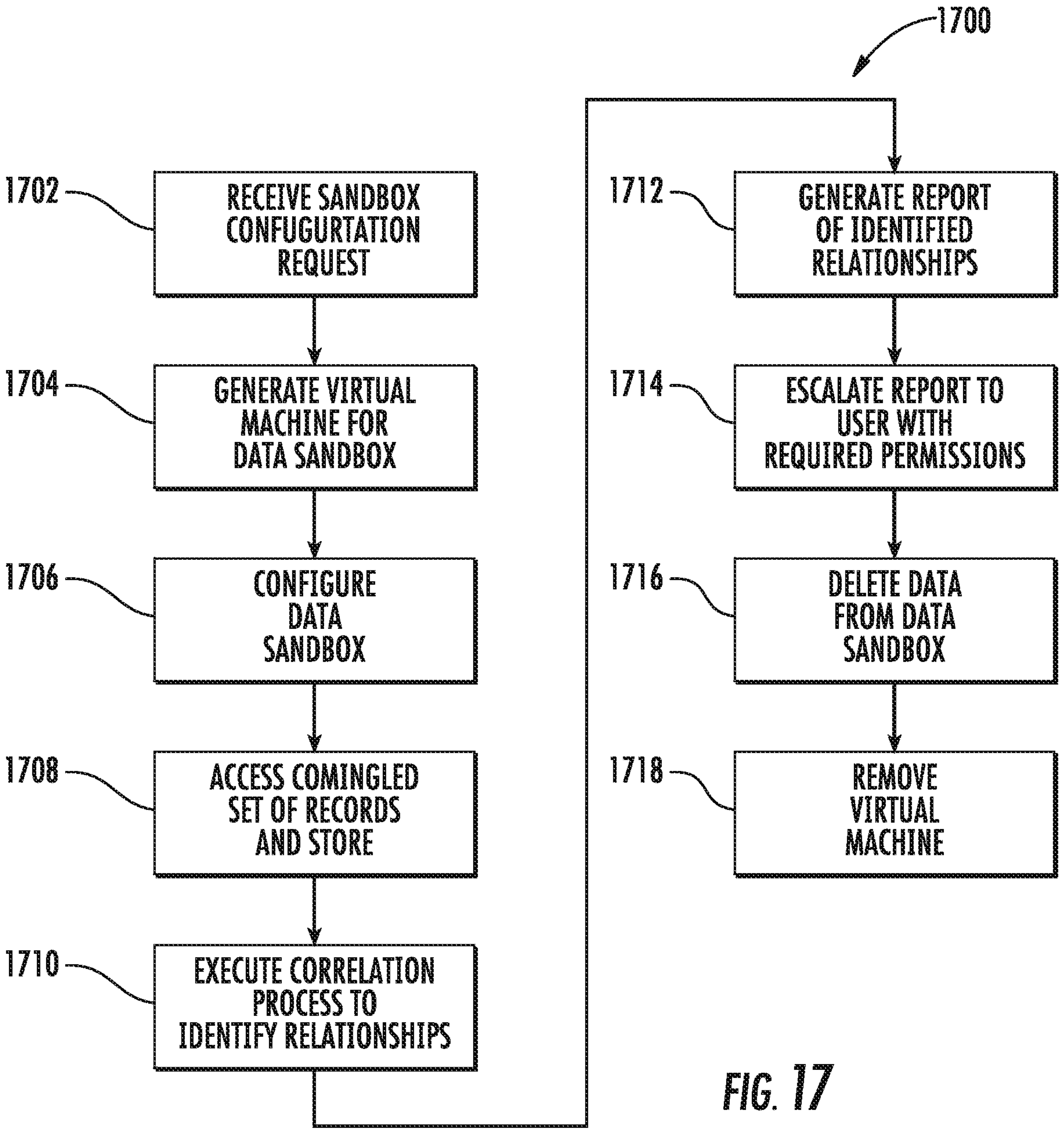
1700
1702
RECEIVE SANDBOX CONFUGURTATION REQUEST
1704
GENERATE VIRTUAL MACHINE FOR DATA SANDBOX
1706
CONFIGURE DATA SANDBOX
1708
ACCESS COMINGLED SET OF RECORDS AND STORE
1710
EXECUTE CORRELATION PROCESS TO IDENTIFY RELATIONSHIPS
1712
GENERATE REPORT OF IDENTIFIED RELATIONSHIPS
1714
ESCALATE REPORT TO USER WITH REQUIRED PERMISSIONS
1716
DELETE DATA FROM DATA SANDBOX
1718
REMOVE VIRTUAL MACHINE

FIG. 17

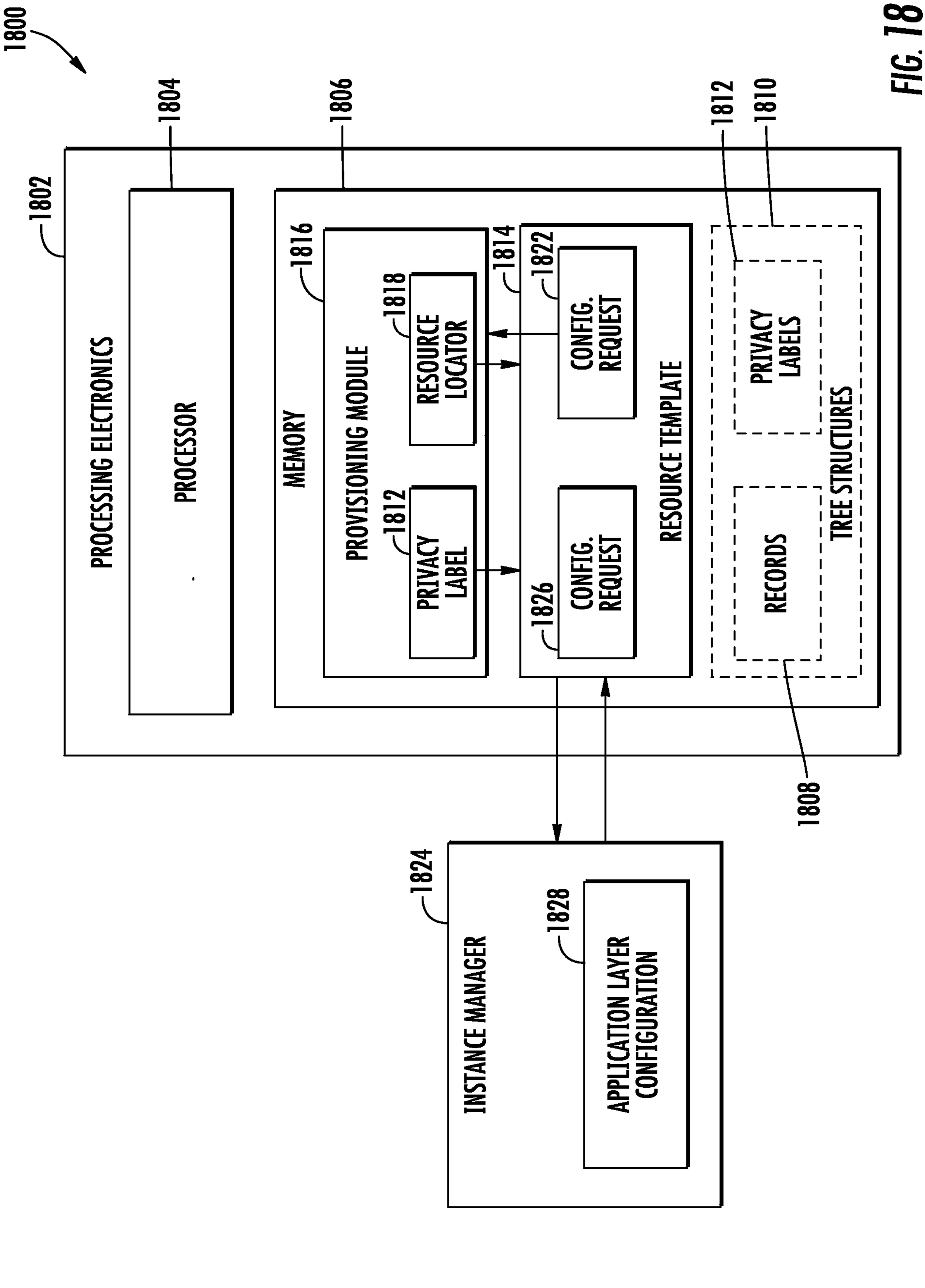
1800
1802
PROCESSING ELECTRONICS
1804
PROCESSOR
1806
MEMORY
1816
PROVISIONING MODULE
1818
RESOURCE LOCATOR
1812
PRIVACY LABEL
1814
RESOURCE TEMPLATE
1822
CONFIG. REQUEST
1826
CONFIG. REQUEST
1812
1810
PRIVACY LABELS
RECORDS
TREE STRUCTURES
1808
1824
INSTANCE MANAGER
1828
APPLICATION LAYER CONFIGURATION

FIG. 18

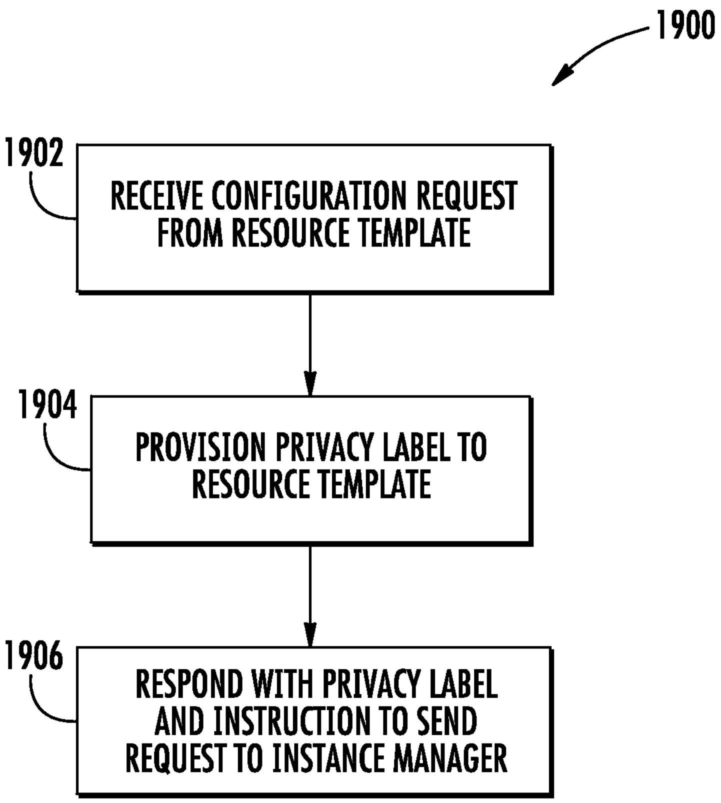
1900
1902
RECEIVE CONFIGURATION REQUEST FROM RESOURCE TEMPLATE
1904
PROVISION PRIVACY LABEL TO RESOURCE TEMPLATE
1906
RESPOND WITH PRIVACY LABEL AND INSTRUCTION TO SEND REQUEST TO INSTANCE MANAGER

FIG. 19

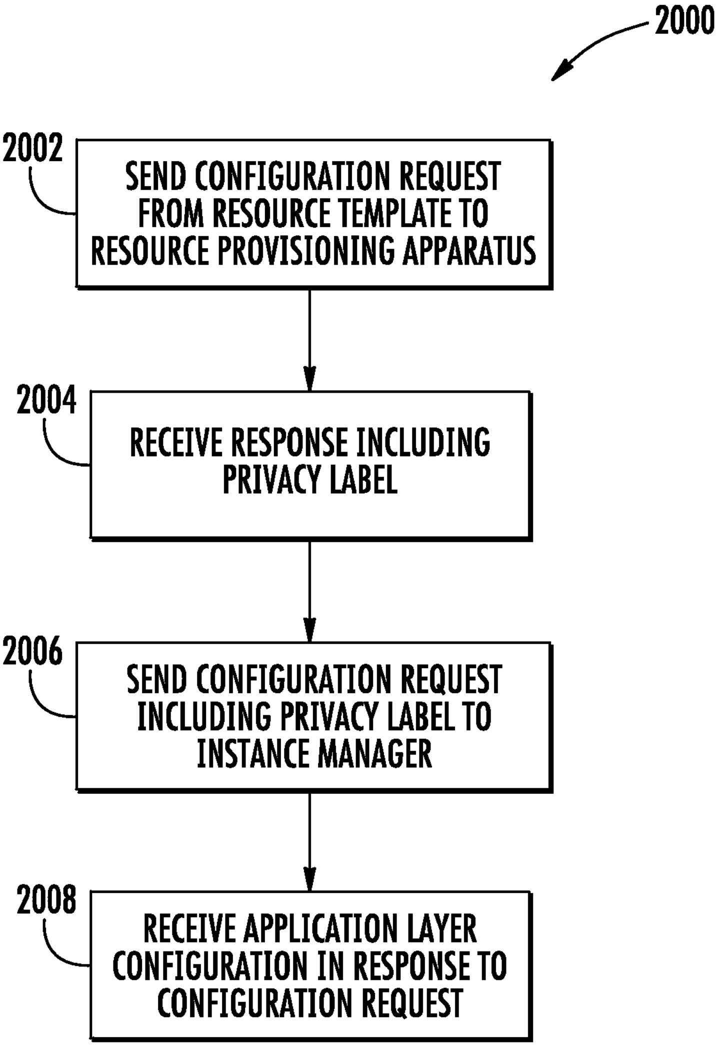
2000
2002
SEND CONFIGURATION REQUEST FROM RESOURCE TEMPLATE TO RESOURCE PROVISIONING APPARATUS
2004
RECEIVE RESPONSE INCLUDING PRIVACY LABEL
2006
SEND CONFIGURATION REQUEST INCLUDING PRIVACY LABEL TO INSTANCE MANAGER
2008
RECEIVE APPLICATION LAYER CONFIGURATION IN RESPONSE TO CONFIGURATION REQUEST

FIG. 20

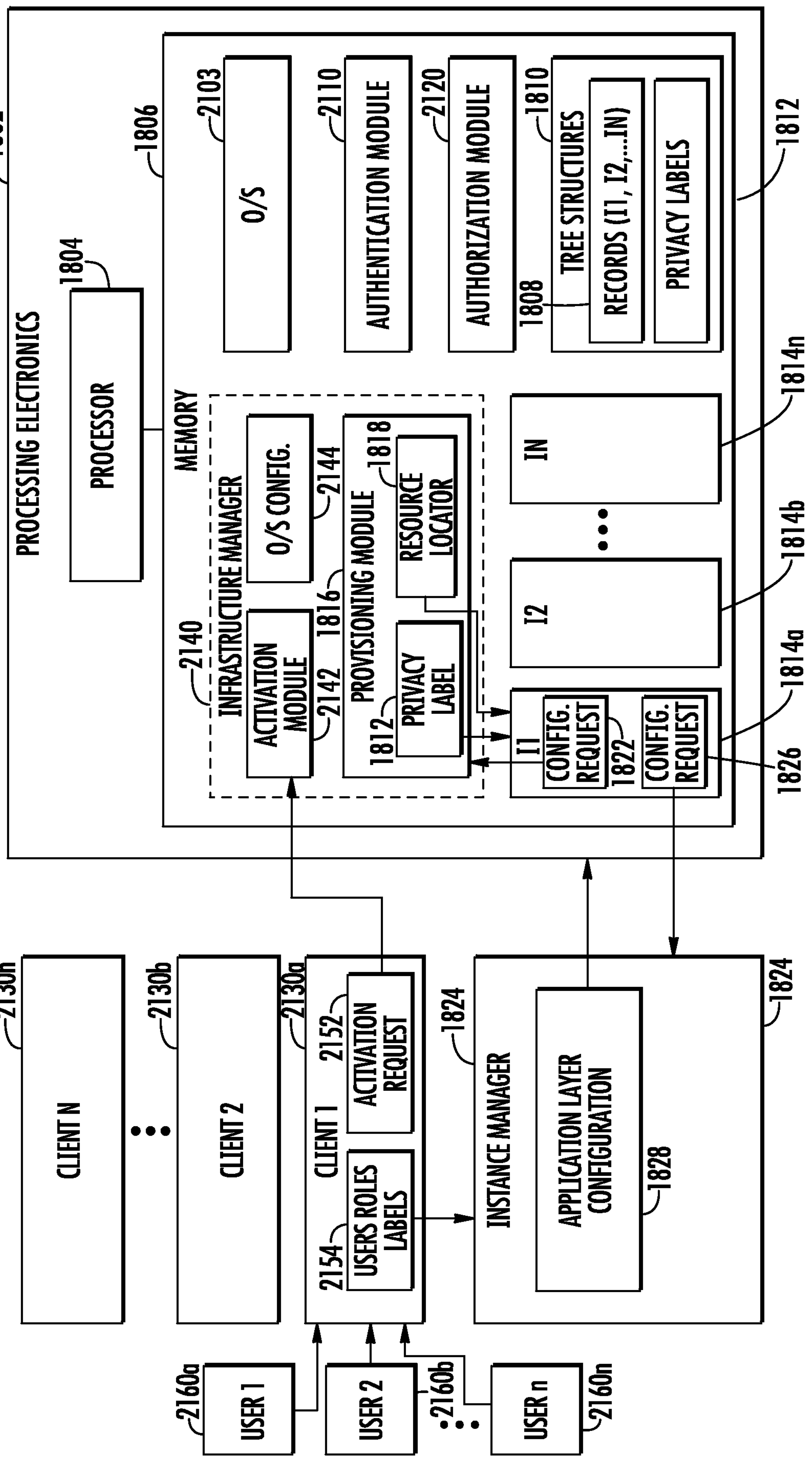
2100
PROCESSING ELECTRONICS
1802
PROCESSOR
1804
MEMORY
1806
O/S
2103
AUTHENTICATION MODULE
2110
AUTHORIZATION MODULE
2120
TREE STRUCTURES
1810
RECORDS (I1, I2,...IN)
1808
PRIVACY LABELS
1812
INFRASTRUCTURE MANAGER
2140
ACTIVATION MODULE
2142
O/S CONFIG.
2144
PROVISIONING MODULE
1816
PRIVACY LABEL
1812
RESOURCE LOCATOR
1818
I1
CONFIG. REQUEST
1822
CONFIG. REQUEST
1826
1814a
I2
1814b
IN
1814n
CLIENT 1
2130a
ACTIVATION REQUEST
2152
USERS ROLES LABELS
2154
CLIENT 2
2130b
CLIENT N
2130n
INSTANCE MANAGER
1824
1824
APPLICATON LAYER CONFIGURATION
1828
USER 1
2160a
USER 2
2160b
USER n
2160n

FIG. 21

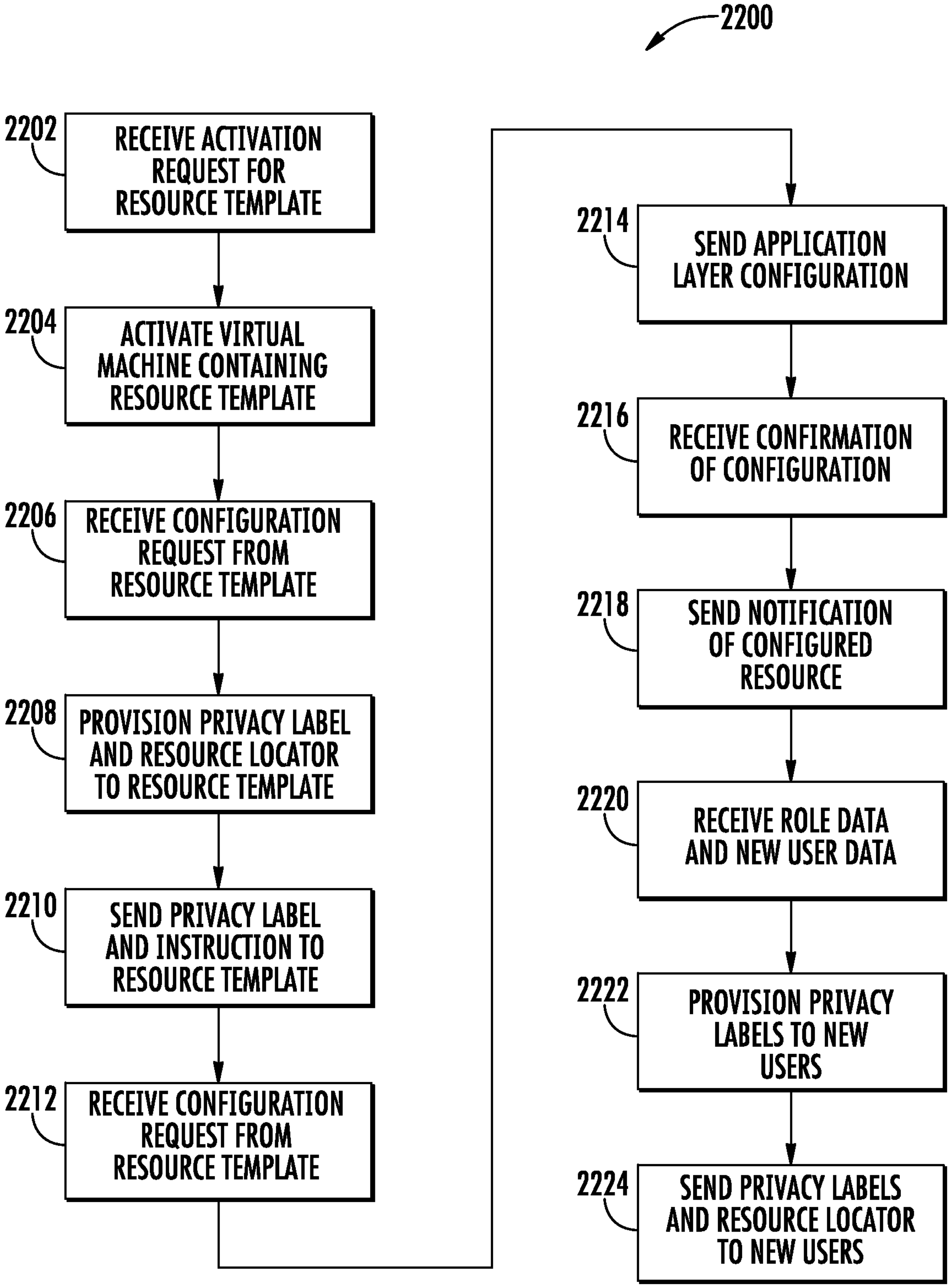
2200
2202
RECEIVE ACTIVATION REQUEST FOR RESOURCE TEMPLATE
2204
ACTIVATE VIRTUAL MACHINE CONTAINING RESOURCE TEMPLATE
2206
RECEIVE CONFIGURATION REQUEST FROM RESOURCE TEMPLATE
2208
PROVISION PRIVACY LABEL AND RESOURCE LOCATOR TO RESOURCE TEMPLATE
2210
SEND PRIVACY LABEL AND INSTRUCTION TO RESOURCE TEMPLATE
2212
RECEIVE CONFIGURATION REQUEST FROM RESOURCE TEMPLATE
2214
SEND APPLICATION LAYER CONFIGURATION
2216
RECEIVE CONFIRMATION OF CONFIGURATION
2218
SEND NOTIFICATION OF CONFIGURED RESOURCE
2220
RECEIVE ROLE DATA AND NEW USER DATA
2222
PROVISION PRIVACY LABELS TO NEW USERS
2224
SEND PRIVACY LABELS AND RESOURCE LOCATOR TO NEW USERS

FIG. 22

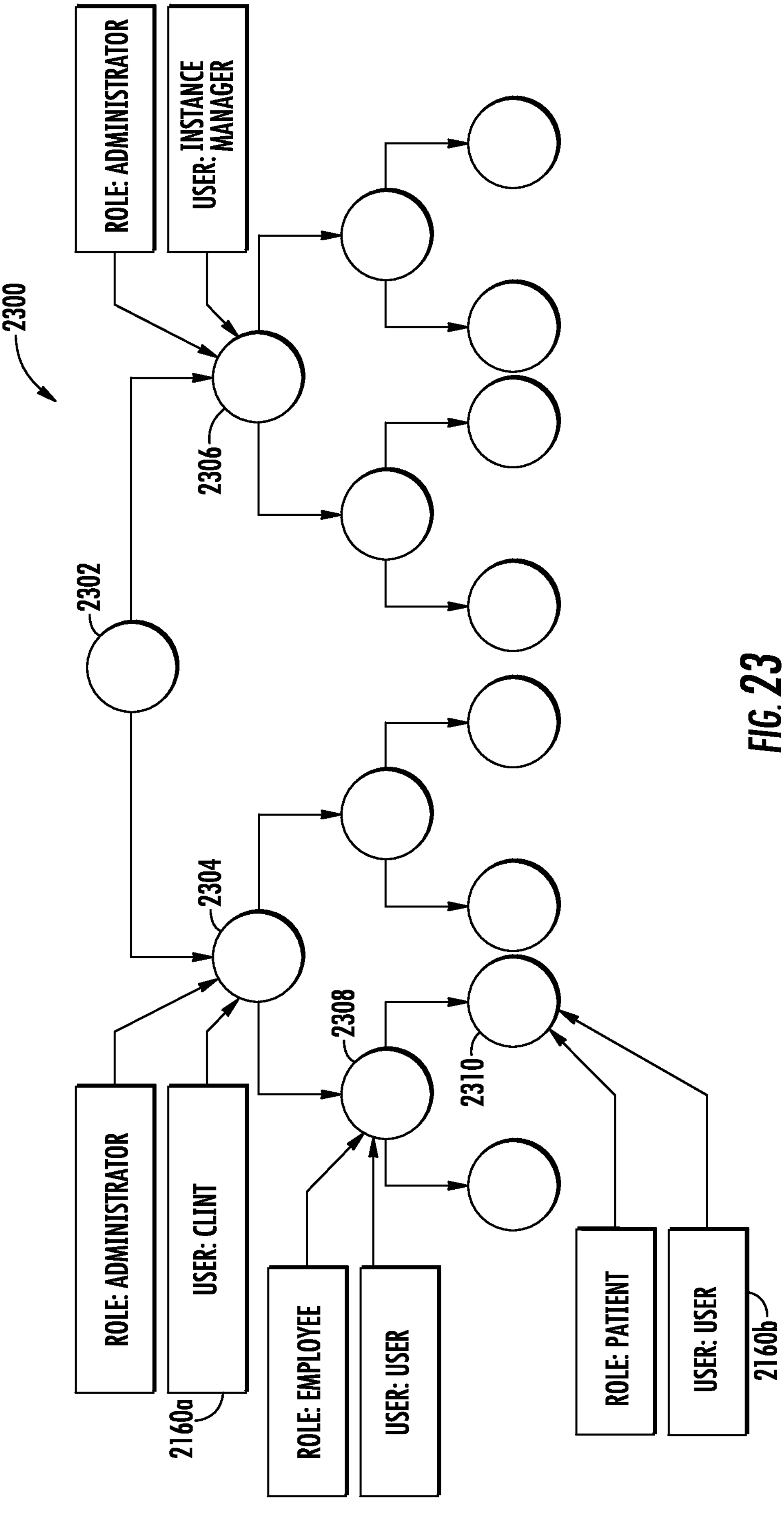
2300
ROLE: ADMINISTRATOR
USER: INSTANCE MANAGER
2306
2302
2304
2308
2310
ROLE: ADMINISTRATOR
USER: CLINT
2160a
ROLE: EMPLOYEE
USER: USER
ROLE: PATIENT
USER: USER
2160b

FIG. 23

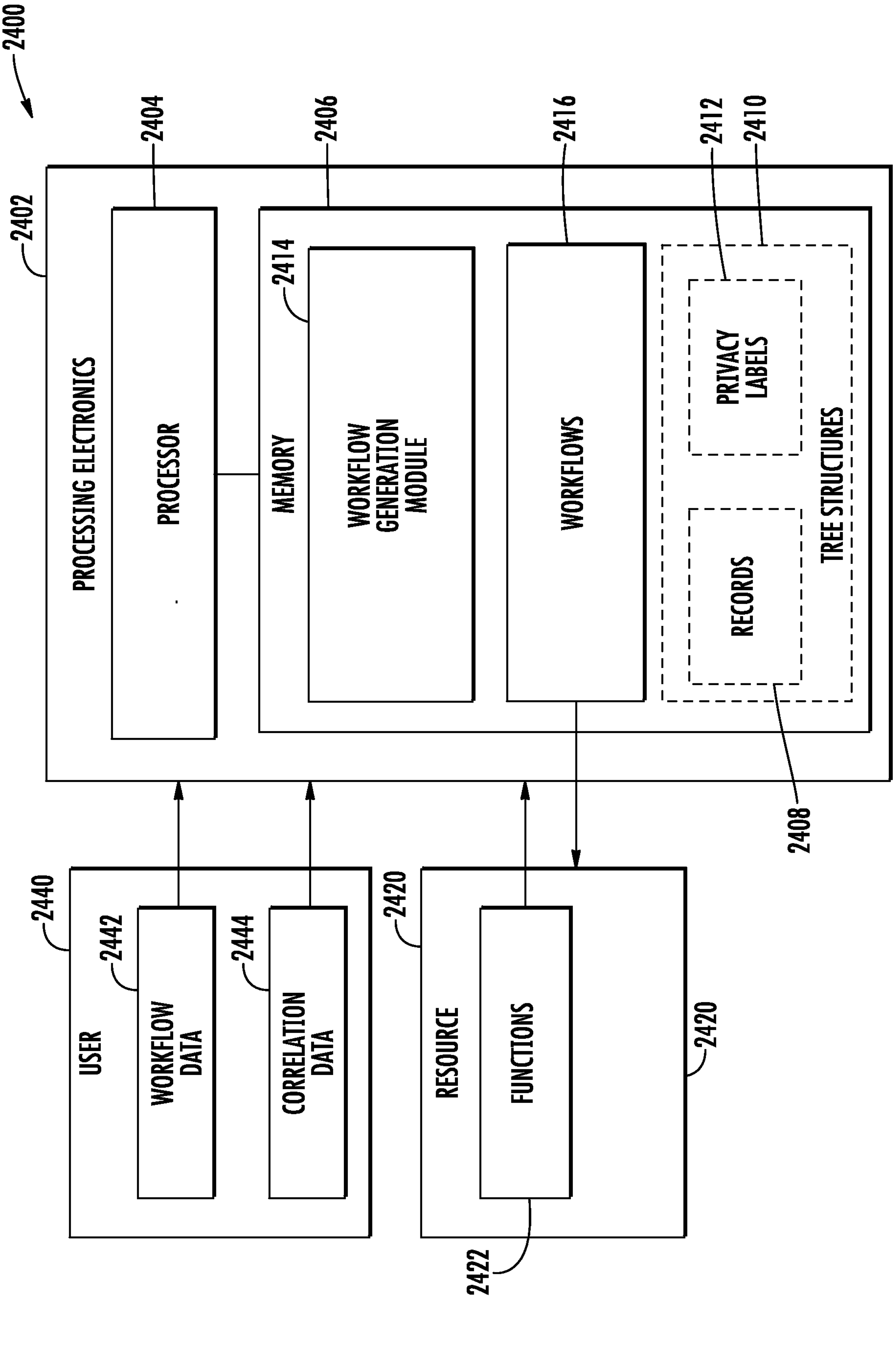
2400
2402
PROCESSING ELECTRONICS
2404
PROCESSOR
2406
MEMORY
2414
WORKFLOW GENERATION MODULE
2416
WORKFLOWS
2412
2410
PRIVACY LABELS
RECORDS
TREE STRUCTURES
2408
2440
USER
2442
WORKFLOW DATA
2444
CORRELATION DATA
2420
RESOURCE
FUNCTIONS
2420
2422

FIG. 24

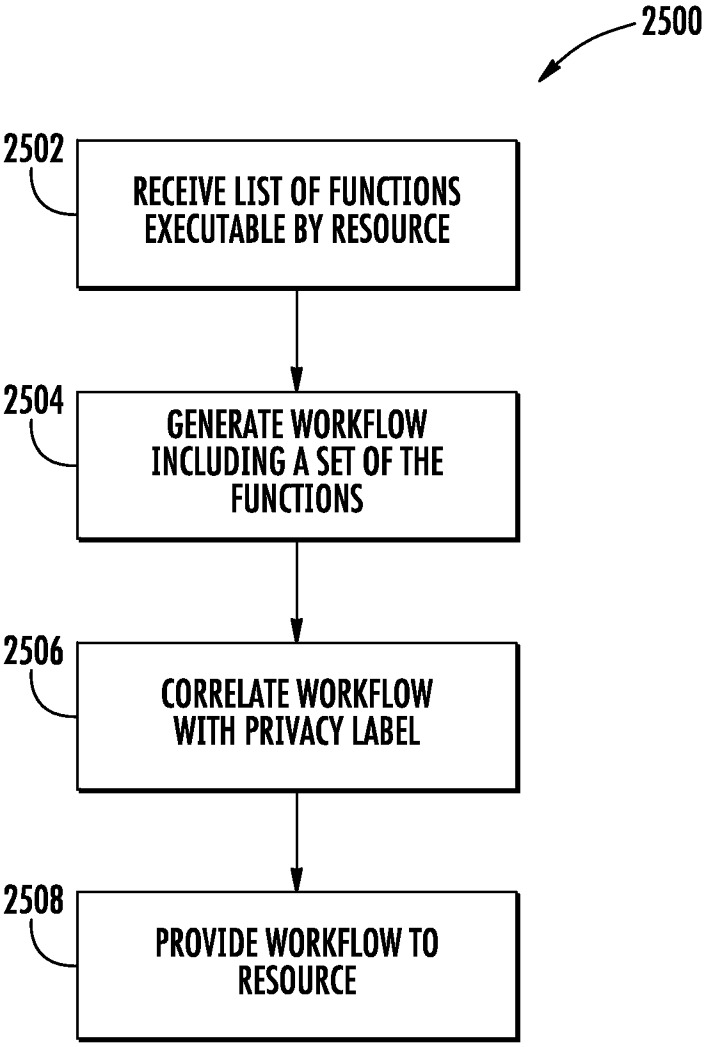
2500
2502
RECEIVE LIST OF FUNCTIONS EXECUTABLE BY RESOURCE
2504
GENERATE WORKFLOW INCLUDING A SET OF THE FUNCTIONS
2506
CORRELATE WORKFLOW WITH PRIVACY LABEL
2508
PROVIDE WORKFLOW TO RESOURCE

FIG. 25

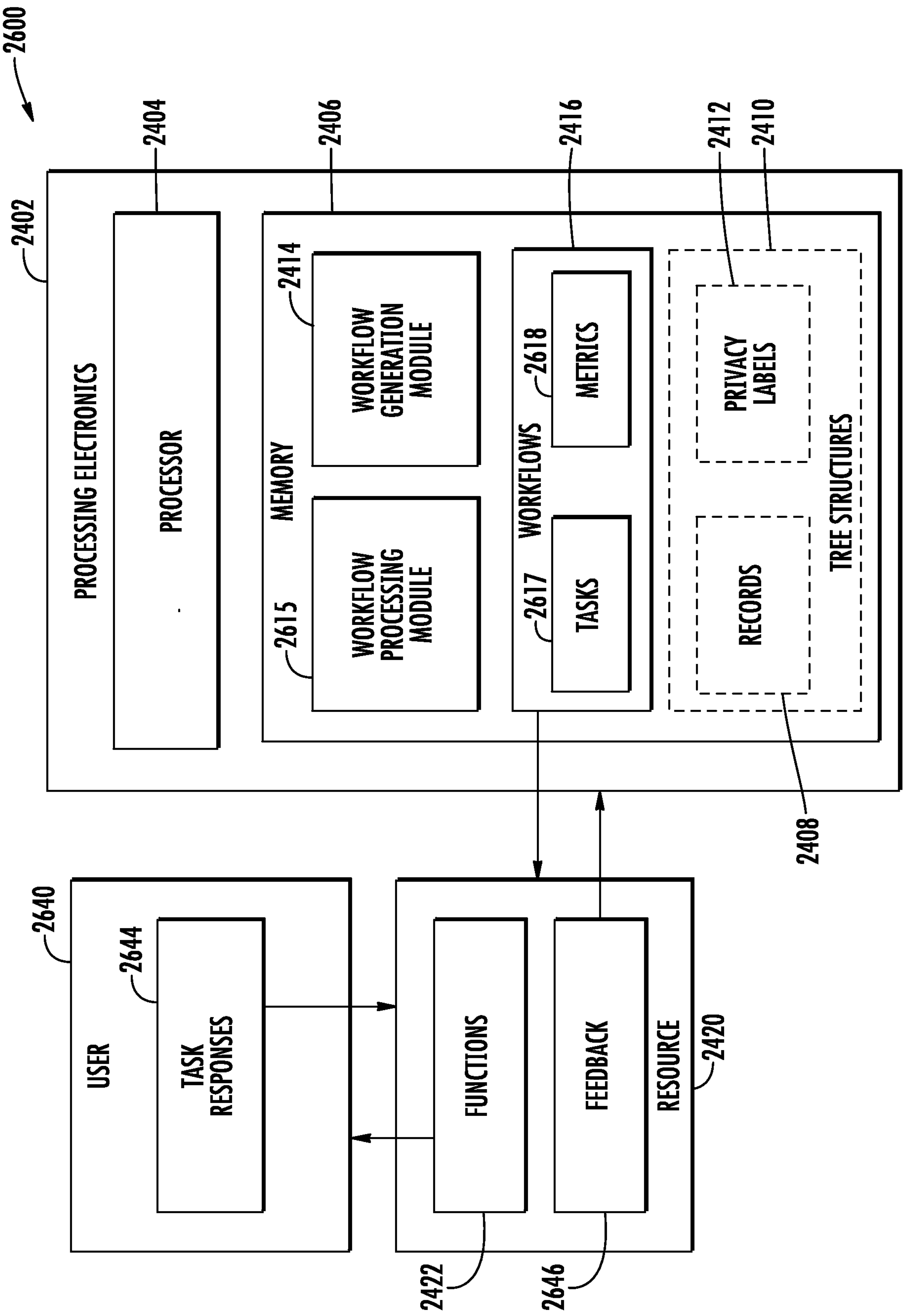
2600
2402
PROCESSING ELECTRONICS
2404
PROCESSOR
2406
MEMORY
2414
WORKFLOW GENERATION MODULE
2615
WORKFLOW PROCESSING MODULE
2416
WORKFLOWS
2618
METRICS
2617
TASKS
2412
2410
PRIVACY LABELS
RECORDS
TREE STRUCTURES
2408
2640
USER
2644
TASK RESPONSES
FUNCTIONS
FEEDBACK
RESOURCE
2420
2422
2646

FIG. 26

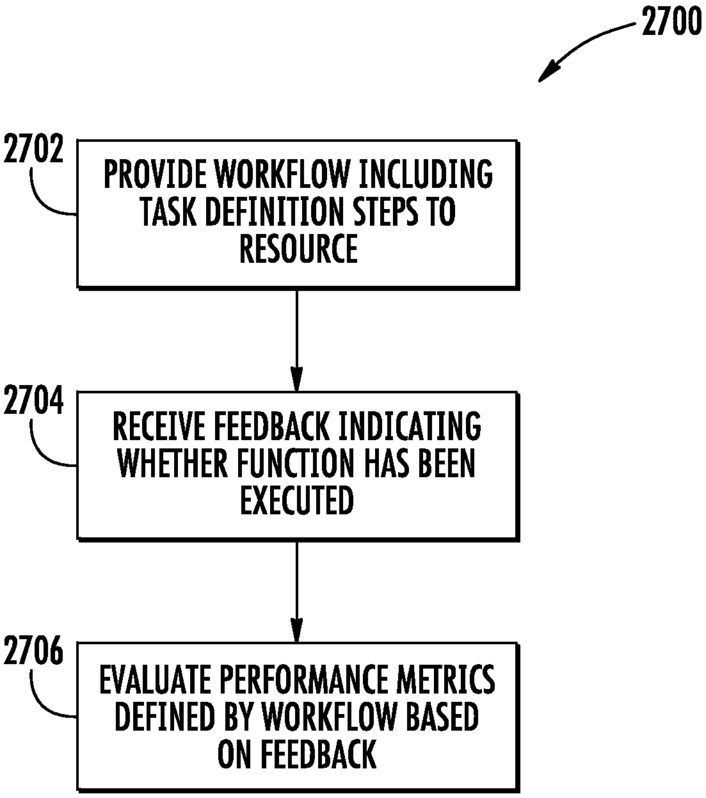
2700
2702
PROVIDE WORKFLOW INCLUDING TASK DEFINITION STEPS TO RESOURCE
2704
RECEIVE FEEDBACK INDICATING WHETHER FUNCTION HAS BEEN EXECUTED
2706
EVALUATE PERFORMANCE METRICS DEFINED BY WORKFLOW BASED ON FEEDBACK

FIG. 27

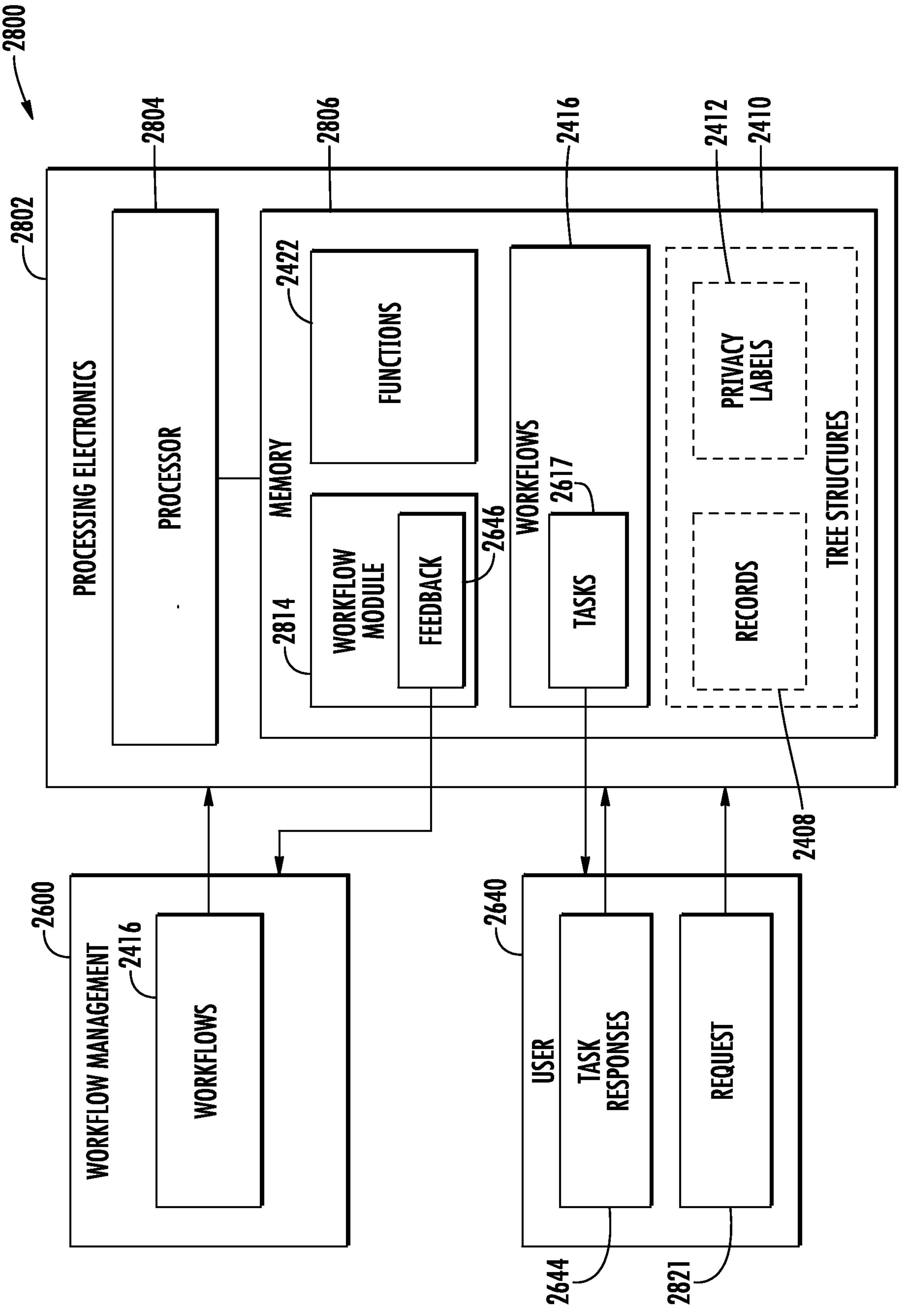
2800
2802
PROCESSING ELECTRONICS
2804
PROCESSOR
2806
MEMORY
2422
FUNCTIONS
2814
WORKFLOW MODULE
FEEDBACK
2646
2416
WORKFLOWS
2617
TASKS
2412
2410
PRIVACY LABELS
RECORDS
TREE STRUCTURES
2408
2600
WORKFLOW MANAGEMENT
2416
WORKFLOWS
2640
USER
TASK RESPONSES
REQUEST
2644
2821

FIG. 28

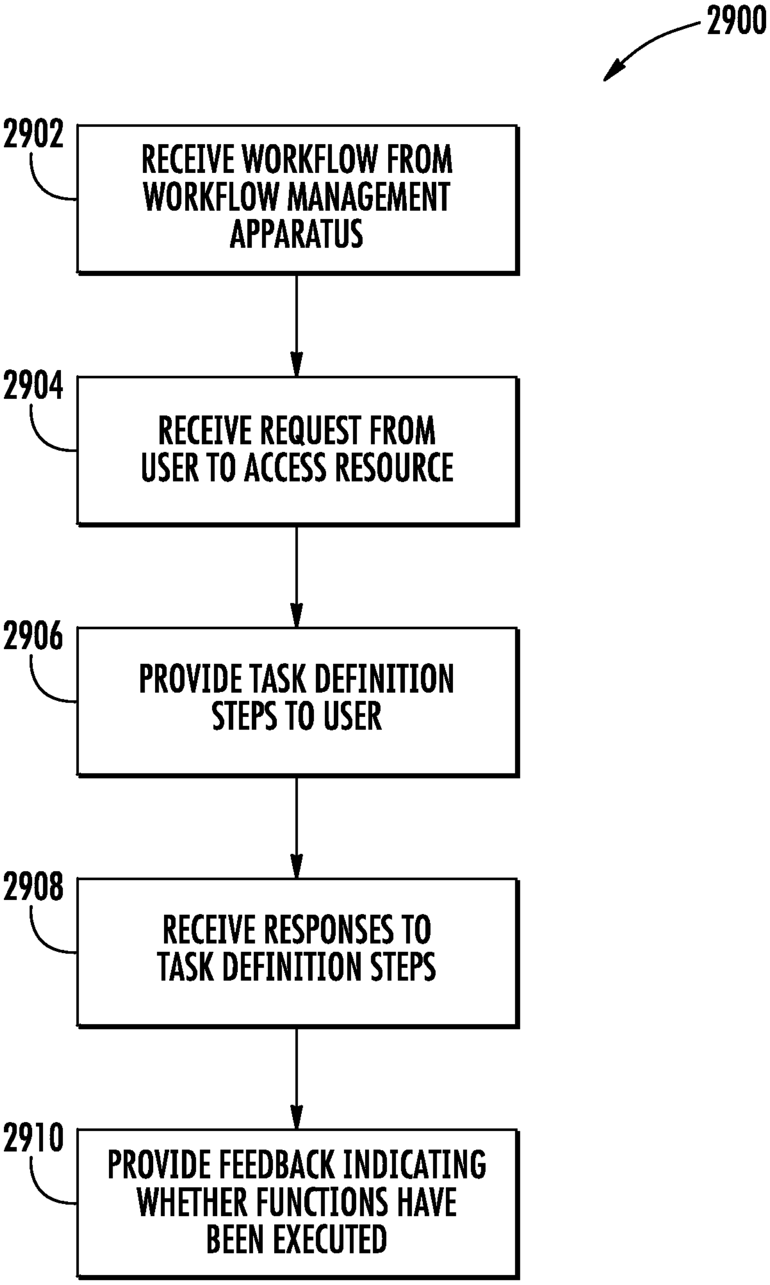
2900
2902
RECEIVE WORKFLOW FROM WORKFLOW MANAGEMENT APPARATUS
2904
RECEIVE REQUEST FROM USER TO ACCESS RESOURCE
2906
PROVIDE TASK DEFINITION STEPS TO USER
2908
RECEIVE RESPONSES TO TASK DEFINITION STEPS
2910
PROVIDE FEEDBACK INDICATING WHETHER FUNCTIONS HAVE BEEN EXECUTED

FIG. 29

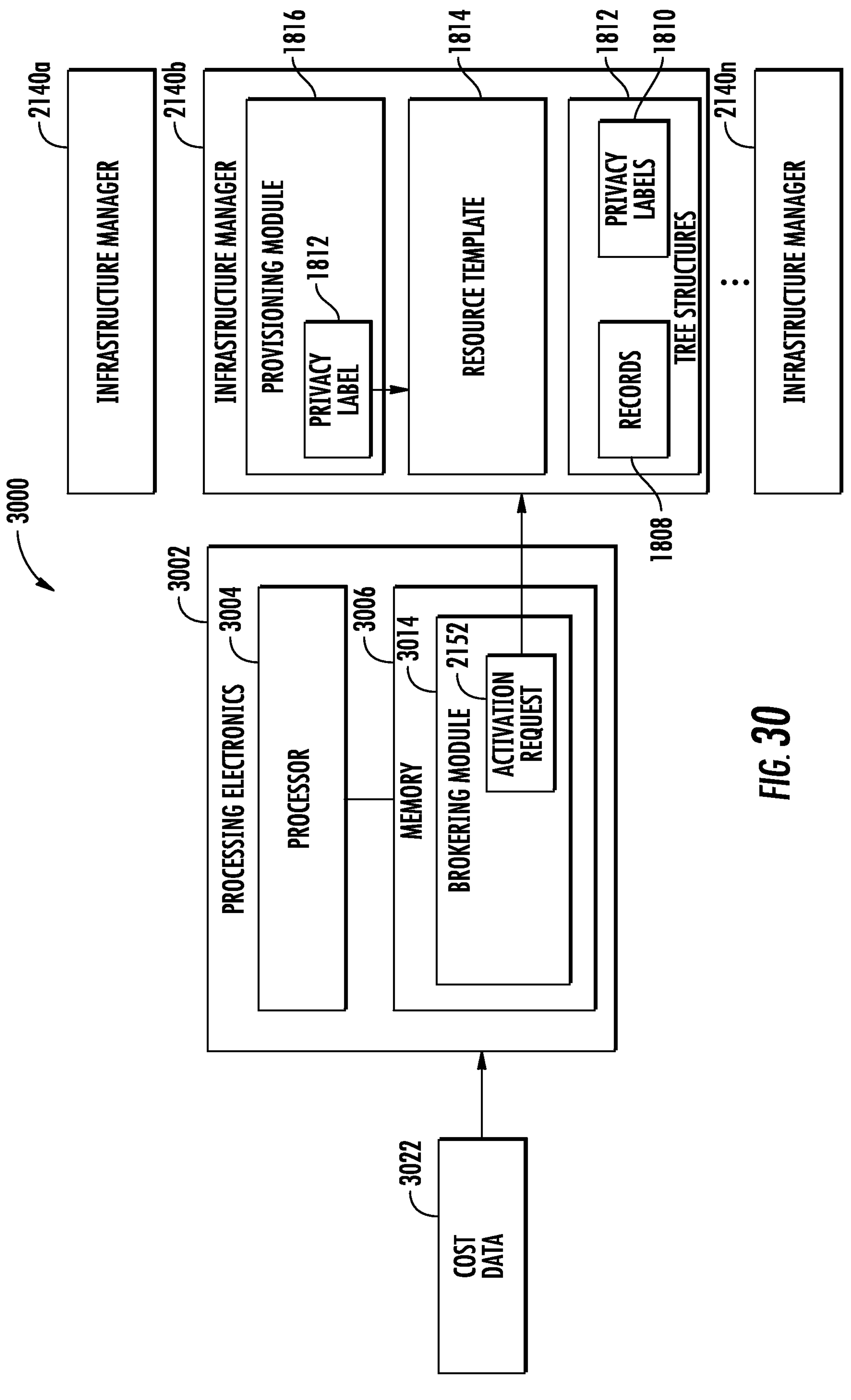
3000
INFRASTRUCTURE MANAGER
2140a
INFRASTRUCTURE MANAGER
2140b
PROVISIONING MODULE
1816
PRIVACY LABEL
1812
RESOURCE TEMPLATE
1814
RECORDS
1808
PRIVACY LABELS
1810
1812
TREE STRUCTURES
INFRASTRUCTURE MANAGER
2140n
PROCESSING ELECTRONICS
3002
PROCESSOR
3004
MEMORY
3006
BROKERING MODULE
3014
ACTIVATION REQUEST
2152
COST DATA
3022

FIG. 30

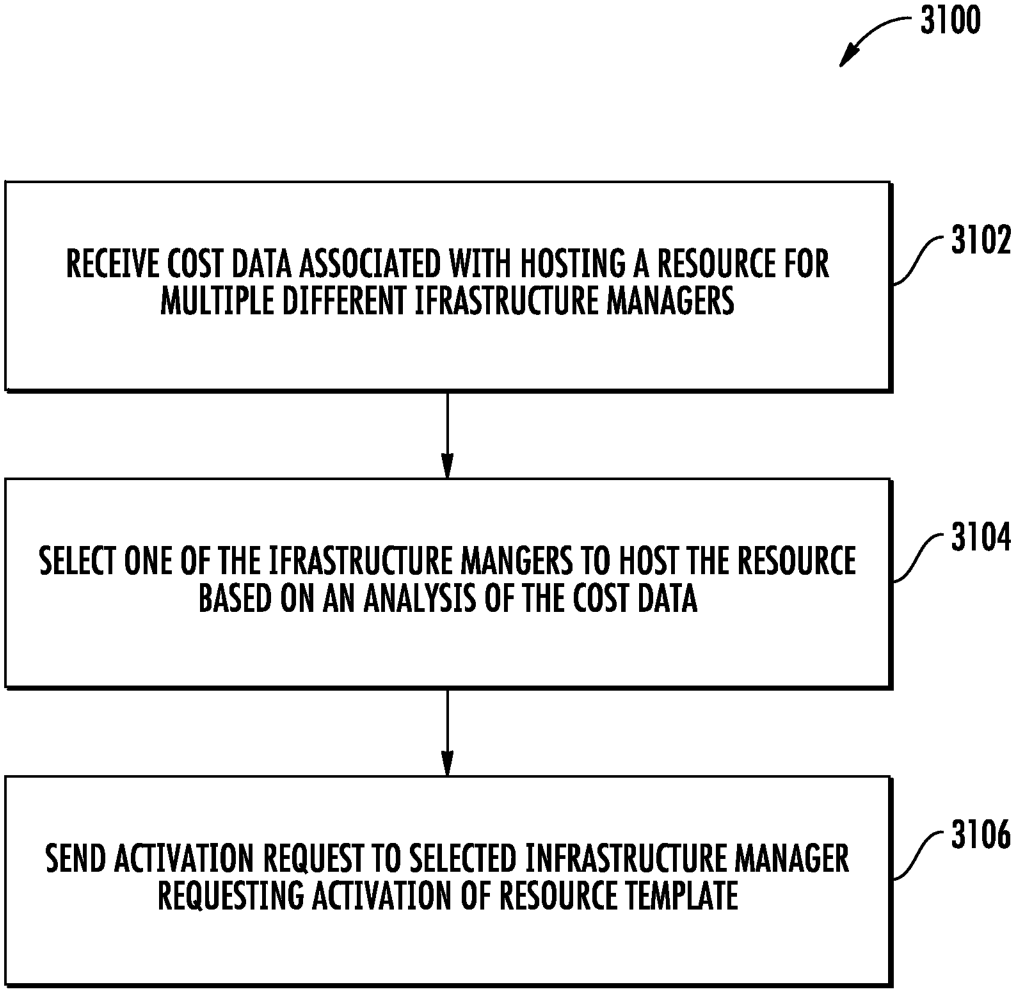
3100
RECEIVE COST DATA ASSOCIATED WITH HOSTING A RESOURCE FOR MULTIPLE DIFFERENT IFRASTRUCTURE MANAGERS
3102
SELECT ONE OF THE IFRASTRUCTURE MANGERS TO HOST THE RESOURCE BASED ON AN ANALYSIS OF THE COST DATA
3104
SEND ACTIVATION REQUEST TO SELECTED INFRASTRUCTURE MANAGER REQUESTING ACTIVATION OF RESOURCE TEMPLATE
3106

FIG. 31

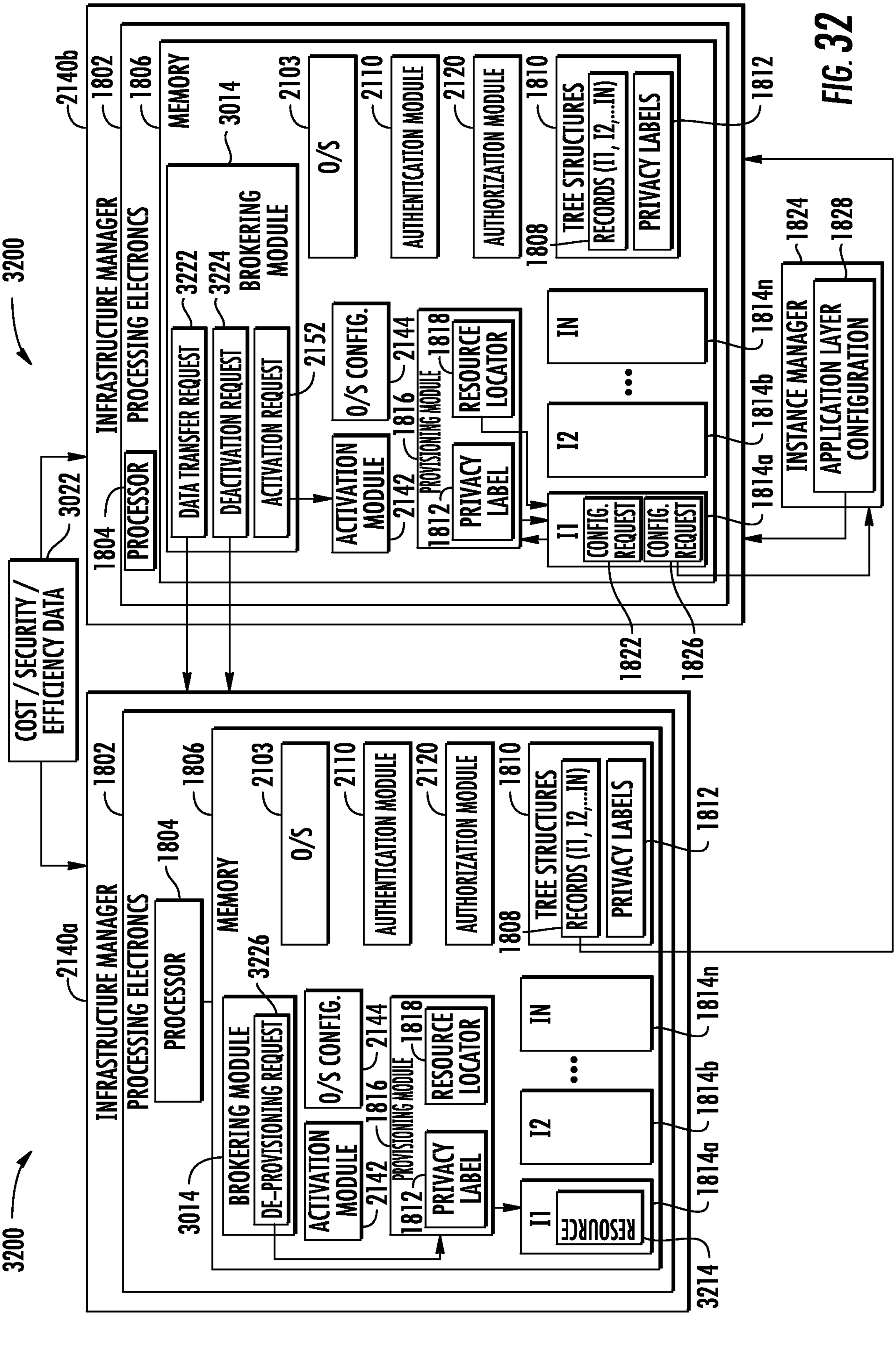
3200
3200
COST / SECURITY / EFFICIENCY DATA
3022
2140b
1802
1806
INFRASTRUCTURE MANAGER
PROCESSING ELECTRONCS
MEMORY
1804
PROCESSOR
3014
BROKERING MODULE
3222
3224
DATA TRANSFER REQUEST
DEACTIVATION REQUEST
ACTIVATION REQUEST
2152
ACTIVATION MODULE
2142
2103
O/S
O/S CONFIG.
2144
1816
PROVISIONING MODULE
1818
RESOURCE LOCATOR
1812
PRIVACY LABEL
2110
AUTHENTICATION MODULE
2120
AUTHORIZATION MODULE
1810
TREE STRUCTURES
1808
RECORDS (I1, I2,...IN)
PRIVACY LABELS
1812
IN
I2
I1
CONFIG. REQUEST
CONFIG. REQUEST
1814n
1814b
1814a
1822
1826
1824
INSTANCE MANAGER
1828
APPLICATION LAYER CONFIGURATION
2140a
1802
1806
INFRASTRUCTURE MANAGER
PROCESSING ELECTRONCS
1804
PROCESSOR
MEMORY
3226
3014
BROKERING MODULE
DE-PROVISIONING REQUEST
ACTIVATION MODULE
2142
O/S CONFIG.
2144
1816
PROVISIONING MODULE
1818
RESOURCE LOCATOR
1812
PRIVACY LABEL
2103
O/S
2110
AUTHENTICATION MODULE
2120
AUTHORIZATION MODULE
1810
TREE STRUCTURES
1808
RECORDS (I1, I2,...IN)
PRIVACY LABELS
1812
IN
I2
I1
RESOURCE
1814n
1814b
1814a
3214

FIG. 32

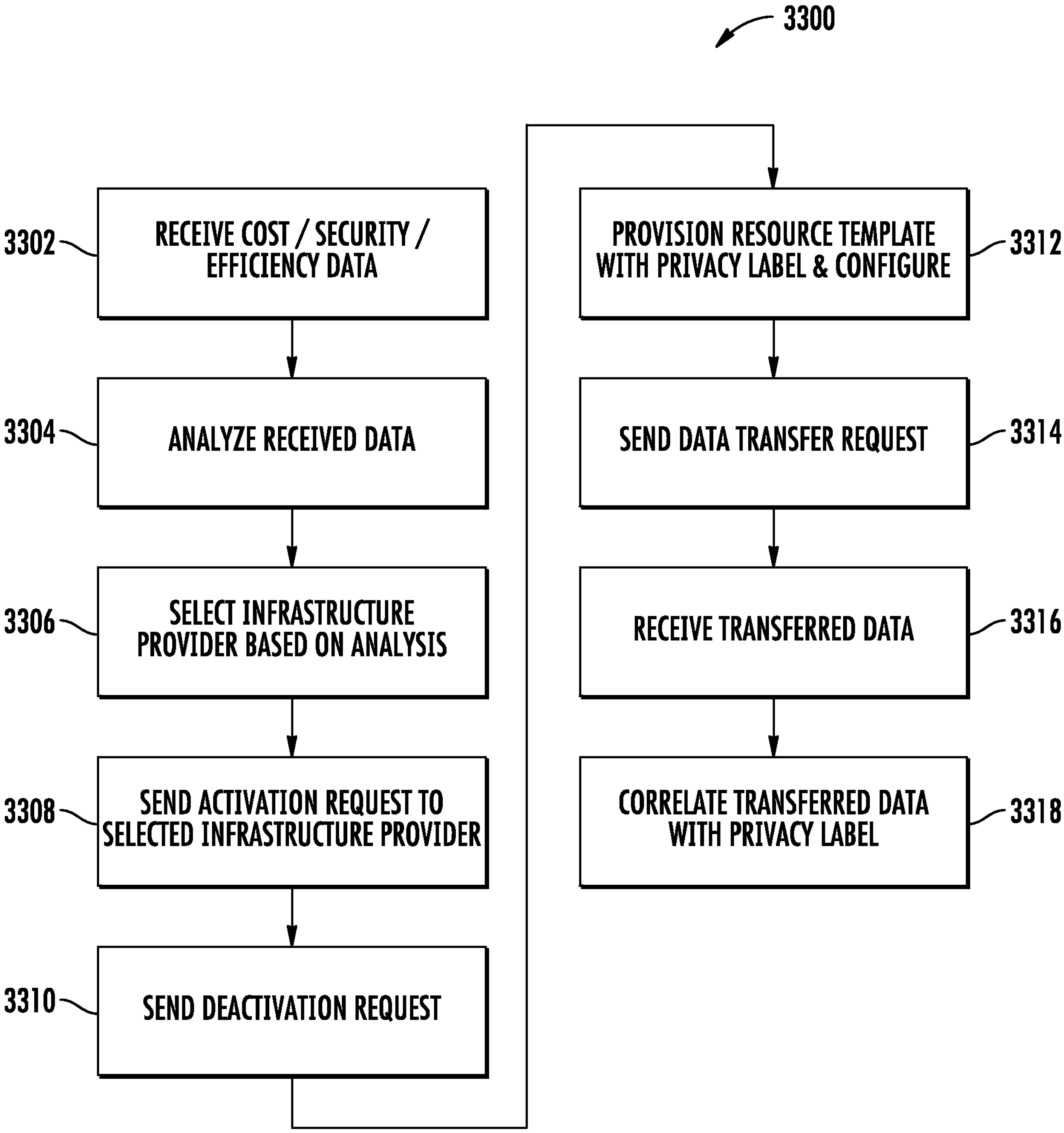
3300
3302
RECEIVE COST / SECURITY / EFFICIENCY DATA
3304
ANALYZE RECEIVED DATA
3306
SELECT INFRASTRUCTURE PROVIDER BASED ON ANALYSIS
3308
SEND ACTIVATION REQUEST TO SELECTED INFRASTRUCTURE PROVIDER
3310
SEND DEACTIVATION REQUEST
3312
PROVISION RESOURCE TEMPLATE WITH PRIVACY LABEL & CONFIGURE
3314
SEND DATA TRANSFER REQUEST
3316
RECEIVE TRANSFERRED DATA
3318
CORRELATE TRANSFERRED DATA WITH PRIVACY LABEL

FIG. 33

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/025768

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/62(2013.01)i, G06F 17/30(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 21/62; G06F 15/173; G06F 9/44; G06F 9/455; G06F 15/177; G06F 9/50; G06F 17/30

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:resource, cost data, manager, multiple server, cloud computing service

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2013-0263117 A1 (RAFAL P. KONIK et al.) 03 October 2013 See paragraphs [0006], [0019]-[0021], [0034], [0068]; and figure 1.	1-15
A	US 2013-0160008 A1 (KEVIN J. CAWLFIELD et al.) 20 June 2013 See paragraphs [0034], [0076], [0085], [0090]; and figure 1.	1-15
A	US 2014-0258446 A1 (CITRIX SYSTEMS, INC.) 11 September 2014 See paragraphs [0080]-[0081]; and figure 6.	1-15
A	US 2012-0226808 A1 (CHRISTOPHER EDWIN MORGAN) 06 September 2012 See paragraphs [0036]-[0037]; and figure 3.	1-15
A	KR 10-2014-0118030 A (INHA INDUSTRY PARTNERSHIP INSTITUTE) 08 October 2014 See paragraphs [0050]-[0051]; and figure 7.	1-15

☐ 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 another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search
23 February 2016 (23.02.2016)

Date of mailing of the international search report
24 February 2016 (24.02.2016)

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

COMMISSIONER

Telephone No. +82-42-481-3578

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/025768

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013-0263117 A1	03/10/2013	None	
US 2013-0160008 A1	20/06/2013	US 2013-159997 A1 US 8694995 B2 US 8694996 B2	20/06/2013 08/04/2014 08/04/2014
US 2014-0258446 A1	11/09/2014	WO 2014-138206 A1	12/09/2014
US 2012-0226808 A1	06/09/2012	US 8959221 B2	17/02/2015
KR 10-2014-0118030 A	08/10/2014	None	