



(19) **United States**

(12) **Patent Application Publication**  
**Schaefer et al.**

(10) **Pub. No.: US 2015/0269390 A1**

(43) **Pub. Date: Sep. 24, 2015**

(54) **SYSTEM AND METHOD OF ESTABLISHING PERMISSION FOR MULTI-TENANCY STORAGE USING ORGANIZATION MATRICES**

(52) **U.S. Cl.**  
CPC ..... *G06F 21/6218* (2013.01); *G06F 17/30289* (2013.01)

(71) Applicant: **PTC Inc.**, Needham, MA (US)  
(72) Inventors: **John Schaefer**, Ambler, PA (US); **Rick Bullotta**, Phoenixville, PA (US)  
(73) Assignee: **PTC Inc.**, Needham, MA (US)  
(21) Appl. No.: **14/222,152**  
(22) Filed: **Mar. 21, 2014**

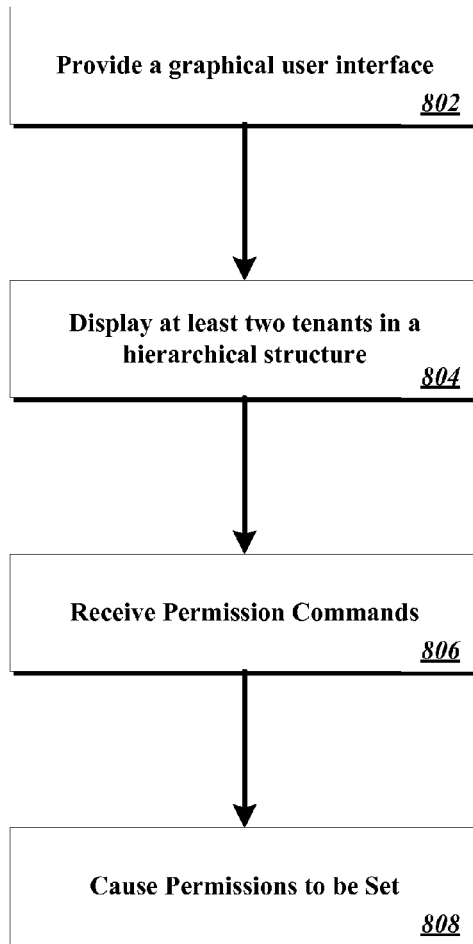
(57) **ABSTRACT**

A system and method of establishing permissions for multi-tenancy storage using an organization matrices is provided. The method includes a computing application for setting permissions for accessing data entries of a multi-tenant database, the data entries storing data associated with a number of computing devices where the multi-tenant database is accessible by two or more tenants and/or sub-tenants. The method includes displaying via a graphical user interface that displays the one or more tenants in a single workspace where the tenants are displayed in a hierarchical structure. The method includes receiving a permission commands to establish the permissions for a tenant and causing the permissions to be set.

**Publication Classification**

(51) **Int. Cl.**  
*G06F 21/62* (2006.01)  
*G06F 17/30* (2006.01)

800 ↘



100

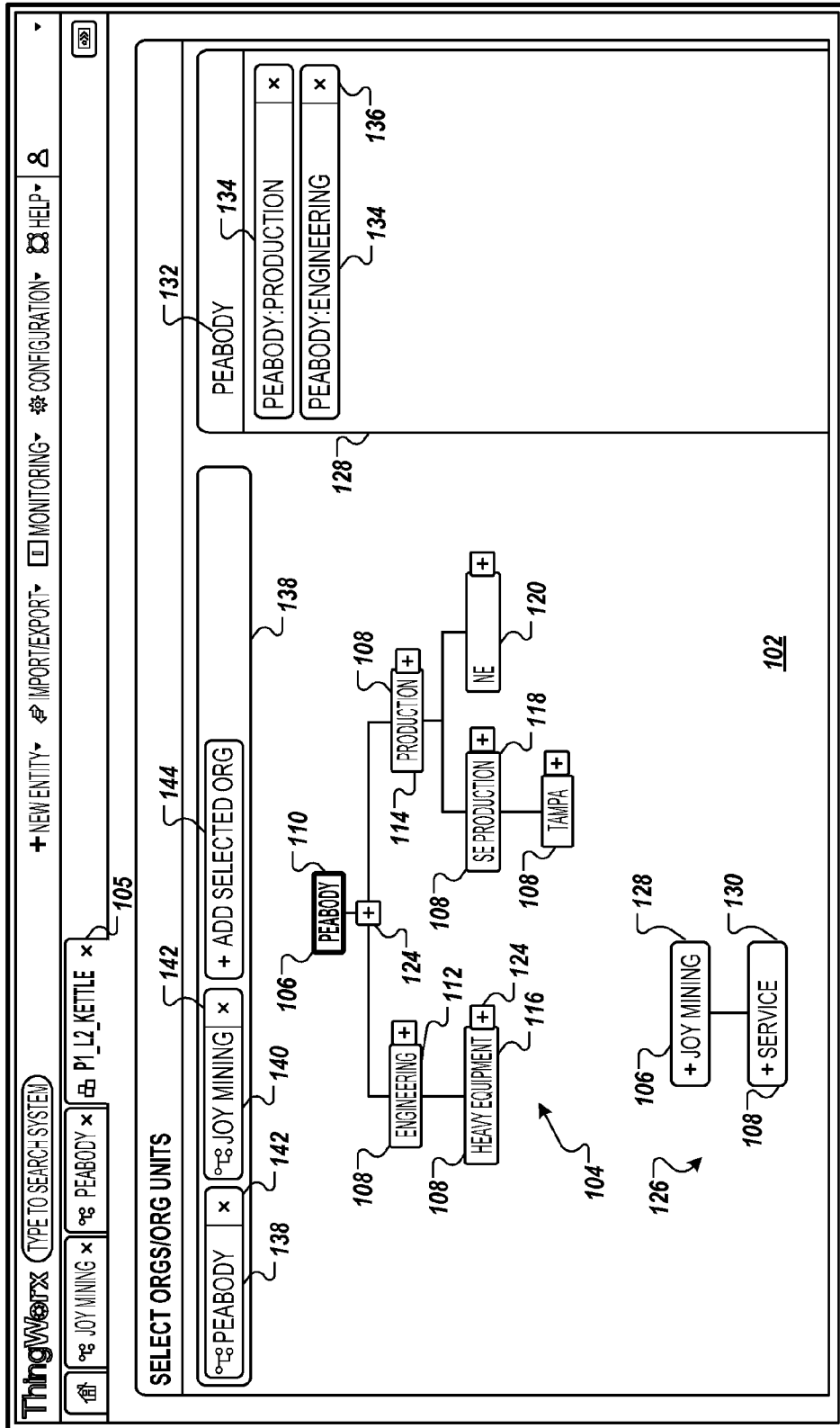


FIG. 1

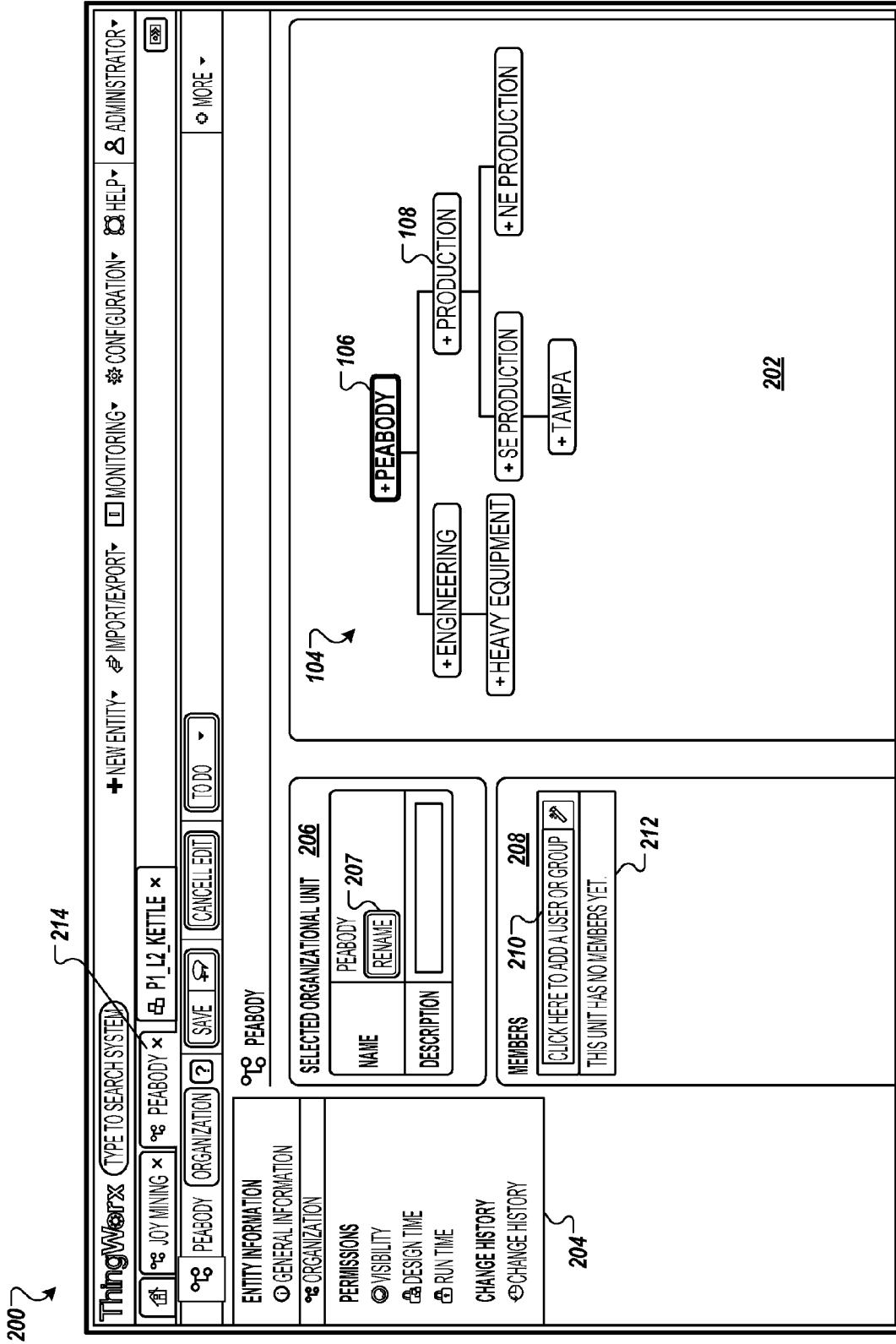


FIG. 2

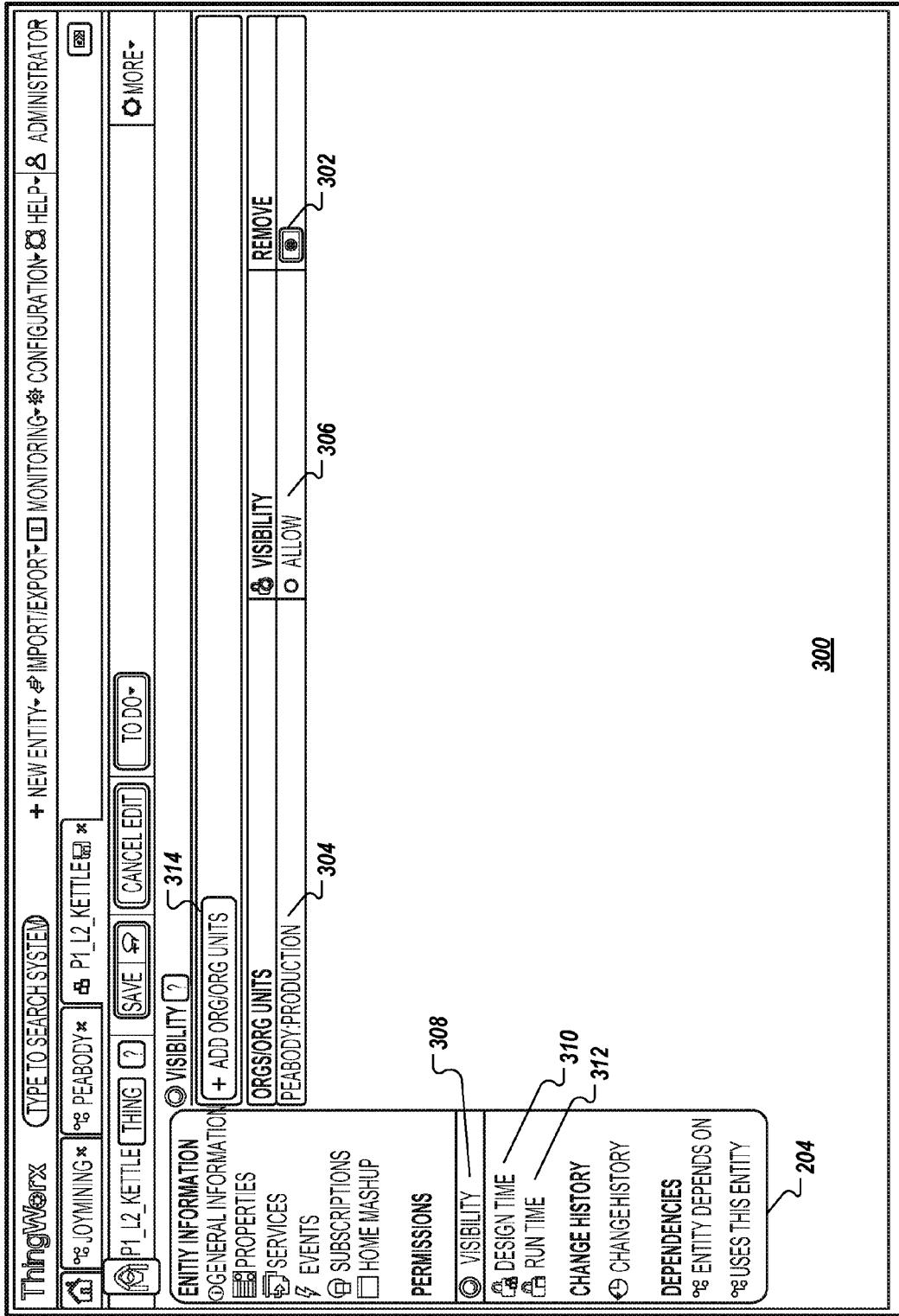


FIG. 3

100

402

424

426

404

406

ThingWorx (TYPE TO SEARCH SYSTEM)

THING PERMISSIONS x

NEW ENTITY IMPORT EXPORT MONITORING CONFIGURATION HELP ADMINISTRATOR

THINGS ? TYPE TO FILTER LIST... ADVANCED CLEAR

+ NEW VIEW EDIT DUPLICATE DELETE PERMISSIONS

FILTERING BY: EXCLUDE SYSTEM OBJECTS 404

NEW	NAME	DESCRIPTION	MODIFIED
<input type="checkbox"/>	ABTSTREHOTETUNELINGTHING	ABTSTREHOTETUNELINGTHING	2014-02-17 13:33:51
<input type="checkbox"/>	SECURITYMONITOR	SECURITYMONITOR	2014-02-17 13:02:47
<input type="checkbox"/>	VAH007_GENERICRTHING	VAH007_GENERICRTHING	2014-02-17 13:02:47
<input type="checkbox"/>	CT_2	CT_2	2014-02-17 13:02:45
<input type="checkbox"/>	VEHICLE42	VEHICLE42	2014-02-17 13:02:45
<input type="checkbox"/>	PL_12_KETTLE	PLANT 1, LINE 2, KETTLE	2014-02-17 13:02:45
<input type="checkbox"/>	STATIONLOC1	KINLER UNIT 1	2014-02-17 13:02:45
<input type="checkbox"/>	PLANT_5	YENGCHANG CHINA PLANT	2014-02-17 13:02:45

THING TEMPLATES  
 THING SHAPES  
 DATA SHAPES  
 NETWORKS  
 MODEL TAGS

DATA STORAGE

PERMISSIONS

VISIBILITY  
 DESIGN TIME  
 RUN TIME

FIG. 4

100 ↗

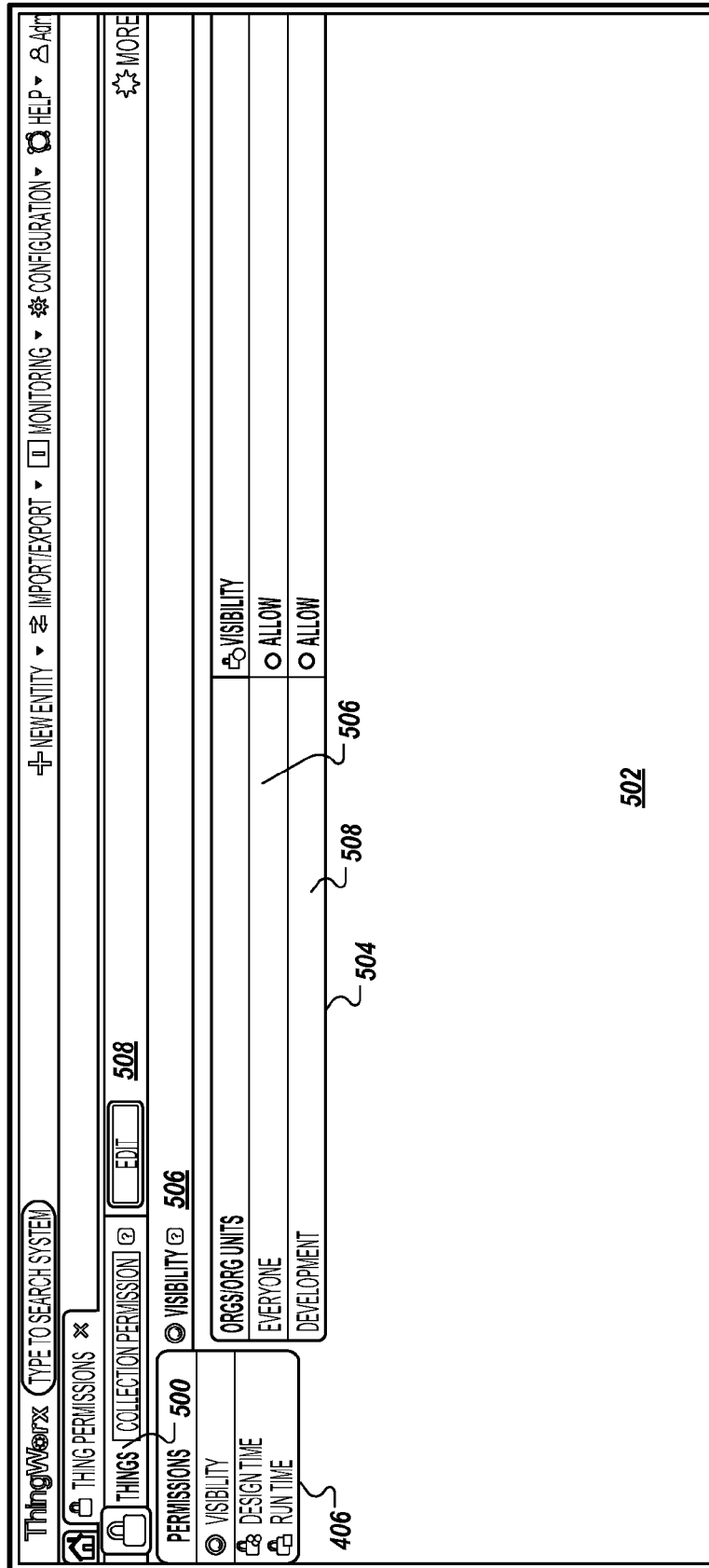


FIG. 5

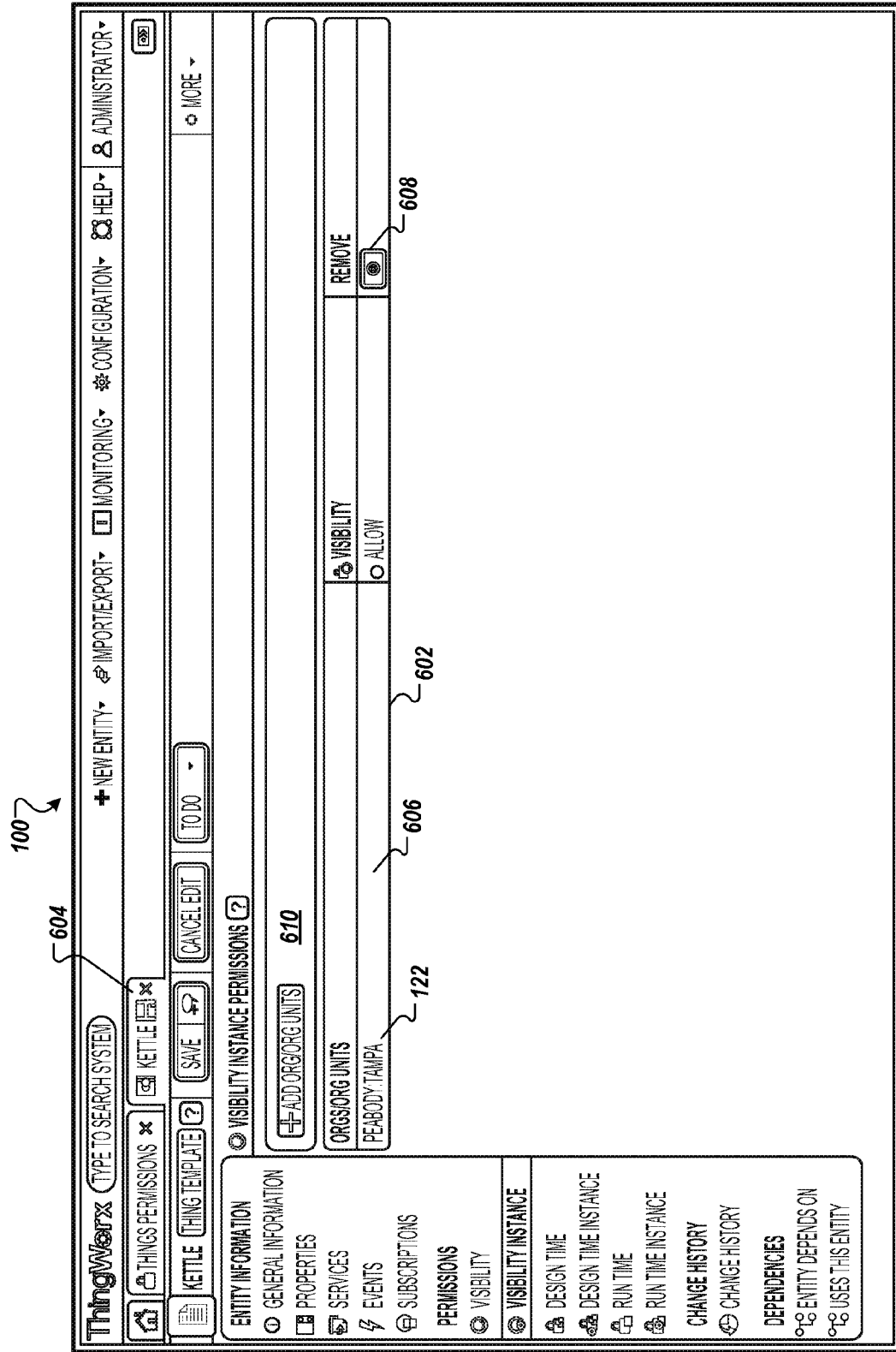


FIG. 6

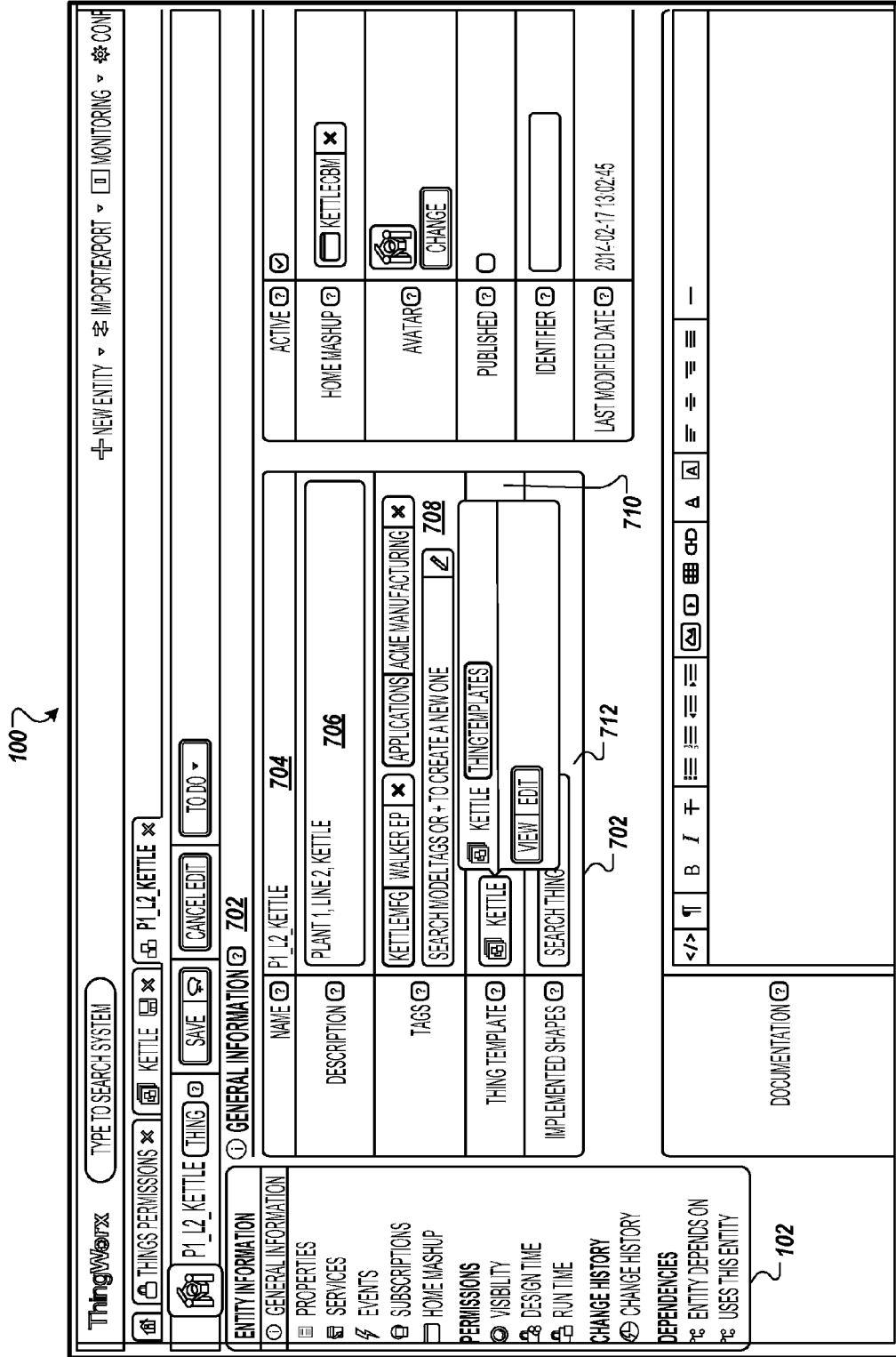
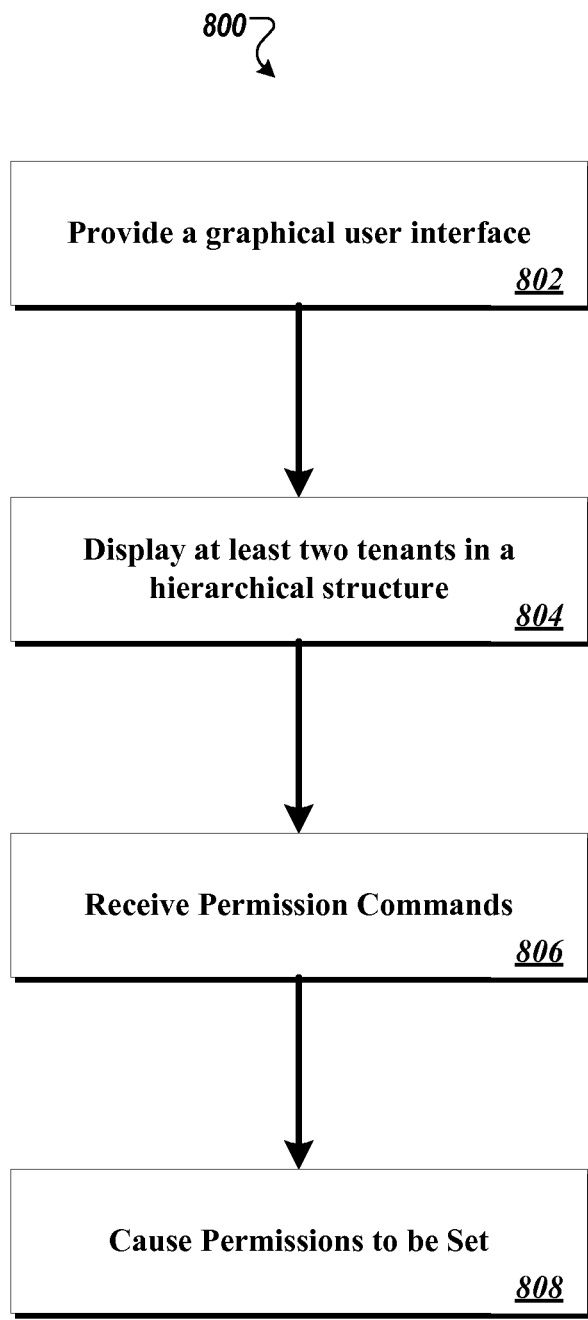


FIG. 7





**FIG. 8**

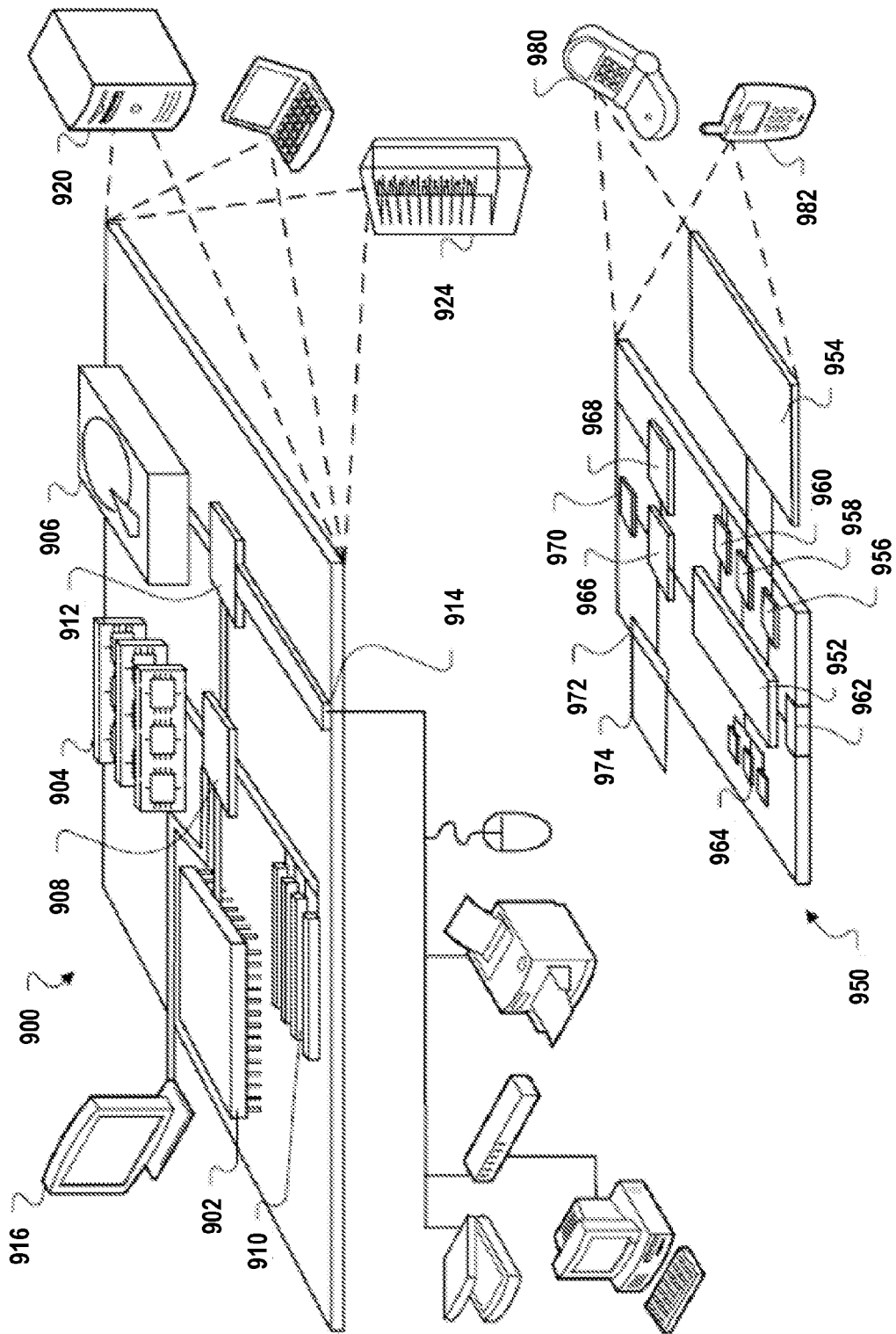


FIG. 9

**SYSTEM AND METHOD OF ESTABLISHING  
PERMISSION FOR MULTI-TENANCY  
STORAGE USING ORGANIZATION  
MATRICES**

FIELD OF THE INVENTION

**[0001]** This invention relates generally to establishing multi-tenancy permissions. More particularly, in certain embodiments, the invention relates to using organization matrices to establish multi-tenancy permissions.

BACKGROUND

**[0002]** Multi-tenant storage generally refers to a computing principle whereby multiple entities, or tenants, occupy and share an instance of a data set running on a set of shared physical or virtual resources. The sharing of the same set of resources offers a given service-provider or owner an opportunity to reduce the operational costs in having less data to manage and less equipment, associated with the data, to maintain.

**[0003]** However, multi-tenant storage, particularly among non-collaborative entities, may be subject by such entities to scrutiny that often adds complexity and oversight that diminishes or exceeds the value proposition in sharing the set of resources. For example, a group of companies may not wish to share a given set of resources if an inadvertent sharing of confidential information exists. The company servicing the data may be required to employ auditing services to ensure that the data is securely shared among the collaborating organizations in a secured manner. To this end, multi-tenant computing is often employed by a single owner to share information and data among subsidiaries or partners.

**[0004]** There exists a need for an approach which reduces the complexity in implementing a secure multi-tenant environment.

SUMMARY

**[0005]** In general overview, an intermediary party provides, to an owner of a multi-tenant database, a computing application to establish multi-tenancy permission to one or more tenants of the database. The computing application, for example, may include a graphical user interface to concurrently establish permissions for the multi-tenant database to one or more tenants and/or sub-tenants.

**[0006]** In some implementations, the data may be associated to industrial equipment, retail equipment, vending machines, transport vehicles, personal computing devices, e.g., cellphones, tablets, and various mobile devices, power generation equipment power distribution equipment, hospital equipment and/or medical devices, computing equipment, rental car fleets, mining equipment, warehouse equipment, among others.

**[0007]** In some implementations, the graphical user interface displays each of the one or more tenants and the one or more sub-tenants in an organization chart. An organization chart is a diagram that shows the structure of members or participants in an organization and relationships among them. An organization chart is a diagram that shows relationships among organizations as it defines whether elements of the organizations are shared or not shared. An organization is made up of a hierarchical set of organization units.

**[0008]** Though organization charts may be subject to change, the inventors realized that specific types of relation-

ships are generally fixed. One such type of relationships is that defined by contracts. Example of contracts may include service contract, purchase and sales contract, and partnership agreement contracts.

**[0009]** To reduce the risk of inadvertent access to data, in some implementations, particularly among non-collaborative entities, the graphical user interface may display the visibility permissions on a single workspace associated to a given device or a class of devices. The graphical user interface may display the visibility permissions as a hierarchical structure, such as an organization chart. This presentation provides additional organization relationship not readily apparent in a list. To this end, an administrator setting the permission can readily identify participants that are incorrectly assigned permission to access data of the given device.

**[0010]** To this end, cross-organizational collaboration defined based on asset may be employed. The collaboration may foster new business models different than traditional multi-tenancy models. Data sharing between an equipment producer and an equipment purchaser, for example, may enable a symbiotic relationship whereby the producer can receive maintenance and usage information about the equipment they sell.

**[0011]** For example, Company A is a manufacturer of a type of industrial equipment. Company A sells its products to ten owner-operated customers, such as "Factories 1 to 10". In a traditional scheme, each Factory owner only has access to the Factory's data, and not that of other factories. As part of an after-purchase service agreement, Company A has been contracted to provide service to the industrial equipment. In being able to provide service from a single database, Company A can analyze and provide blind-benchmark information to some or all of the ten owners. Company A further benefits in having data that it can share among its internal team that may be employed to improve the design and operation of their industrial-equipment products.

**[0012]** To reduce the risk of advertent sharing of confidential information, in some implementations, the computing application allows an administrator to set visibility permission to the multi-tenant database via the graphical user interface. The visibility permission may serve as a global permission for a given data set within the multi-tenant database. The visibility permission may establish whether a participant has any access to a given portion of the multi-tenant database. Subsequent to setting visibility permissions, in some implementations, administrator of the multi-tenant database may set granular levels of permission types.

**[0013]** In addition, cross-organizational collaboration may be defined for a class of assets. To this end, the computing application may establish permissions for multiple assets and for multiple tenants (including sub-tenants) in an organization chart. The multiple asserts may have been defined within the multi-tenant database by a set of common property definitions.

**[0014]** Visibility permission is a form of access control that may be extended to an organization and its members. In some implementations, if data is visible to an organizational unit, then members of the organization may also have access to the data subject to the granular security permissions. The underlying, granular security permissions allows an administrator to specific the interactions (i.e., access) that any member of an organization unit may have with a specific asset (and/or class of assets). Examples of such permission types may include read access, write access, modify access, and change permis-

sions access. If an organization does not have visibility permission, then that data essentially does not exist to that organization or the organization members.

**[0015]** Granular security model can be employed to allow each company access to only the data elements, services, etc. that they are entitled to. For example, Beverage Company A leases a vending machine (VM101) to Operating Company B. Operating Company B outsources the maintenance and inventory of the vending machine to Supplier C. Operating Company B also leases the vending machines from Beverage Company D. Here, Beverage Company A maintains control of the leased vending machine under a maintenance and operation agreement with Company B.

**[0016]** The visibility permission of the data associated to the vending machine "VM101" may be granted to organization units of the three companies (B, C, and D) depending on their contractual relationships with the Company A. Company A, as the administrator, may have all of the levels of permissions (e.g., read, write, modify, and change), including the permission to assign visibility and granular security permissions to other organizations. To view the performance of its leased assets, Company A may grant read-permissions to Company B. To service the inventory of the vending machine, Company A may grant read and write permissions to Company C. To prevent Company D from accessing the data, Company A may not grant visibility permission to Company D.

**[0017]** Applications for the systems and methods described herein are not limited to the aforementioned examples, but may be deployed in any number of contexts, as would be understood by one of ordinary skill in the art. Contents of the background are not to be considered as an admission of the contents as prior art.

**[0018]** In one aspect, the present disclosure describes a method of establishing permissions for multi-tenancy storage using an organizational matrices. The method includes providing, by a processor of a computer, a computing application for setting permissions for accessing data entries of a multi-tenant database. The data entries store data associated with one or more computing devices where the multi-tenant database is accessible by two or more tenants. The tenants may include a first sub-tenant and a second sub-tenant where the sub-tenants are a part of the first tenant. The tenants may include a third sub-tenant and a fourth sub-tenant where these sub-tenants are a part of the second tenant. At least two of the tenants may not be part of the same organization or company.

**[0019]** In some implementations, the method includes displaying, via the computing application, a graphical user interface associated with the computer where the graphical user interface displays the tenants in a single workspace where the tenants are displayed with a hierarchical structure. The graphical user interface may display each of the first tenant and second tenant as an object where the object includes a graphical input to add a sub-tenant to the respective first tenant and second tenant. The graphical user interface may display each of the first tenant and second tenant as an object where the object includes a graphical input to add a sub-tenant to the respective first tenant and second tenant.

**[0020]** In some implementations, the method includes receiving, in the workspace at the graphical user interface of the computing application, a first permission command to establish the permissions for a first tenant and a second permission command to establish the permissions for a second tenant. The graphical user interface may be configured to

receive at least one permission setting selected from group consisting of a read permission, a write permission, and a modify permission.

**[0021]** In some implementations, the method includes causing, by a processor of the computing application, the permissions for the first tenant and the second tenant to be set using the received first permission command and the received second permission command.

**[0022]** In some implementations, the method further includes receiving, in the workspace at the graphical user interface of the computing application, a third permission command to establish the permissions for a third tenant of the tenants where the graphical user interface shows the first object and the second object as hierarchical subunits of the third object.

**[0023]** In one aspect, the present disclosure describes a system including a processor and a memory, the memory storing instruction that, when executed by the processor, cause the processor to provide a computing application for setting permissions for accessing data entries of a multi-tenant database. The data entries stores data associated with one or more computing devices where the multi-tenant database is accessible by two or more tenants. The tenants may include a first sub-tenant and a second sub-tenant where sub-tenants are a part of the first tenant. The tenants may include a third sub-tenant and a fourth sub-tenant where the sub-tenants are a part of the second tenant. At least two of the tenants may not be part of the same organization or company.

**[0024]** In some implementations, the instructions, when executed, further cause the processor to display, via the computing application, a graphical user interface associated with the computer where the graphical user interface displays the tenants in a single workspace, and wherein the tenants are displayed with a hierarchical structure. The graphical user interface may display each of the tenants as an object where the object includes a graphical input to add a sub-tenant to the respective tenants. The graphical user interface may display each of the tenants as an object where the object includes a graphical input to add a sub-tenant to the respective tenants.

**[0025]** In some implementations, the instructions, when executed, further cause the processor to receive, in the workspace at the graphical user interface of the computing application, a first permission command to establish the permissions for a first tenant and a second permission command to establish the permissions for a second tenant. The graphical user interface may be configured to receive at least one permission setting selected from group consisting of a read permission, a write permission, and a modify permission.

**[0026]** In some implementations, the instructions, when executed, further cause the processor to cause, by a processor of the computing application, the permissions for the tenants to be set using the received permission commands.

**[0027]** In some implementations, the instructions, when executed, further cause the processor to receive, in the workspace at the graphical user interface of the computing application, a third permission command to establish the permissions for a third tenant where the graphical user interface shows the first object and the second object as hierarchical subunits of the third object.

**[0028]** In one aspect, the present disclosure describes a non-transitory computer readable medium having instructions stored thereon, where the instructions, when executed by a processor, cause the processor to provide a computing

application for setting permissions for accessing data entries of a multi-tenant database. The data entries stores data associated with a number of computing devices where the multi-tenant database is accessible by two or more tenants. The tenants may include a first sub-tenant and a second sub-tenant where the sub-tenants are a part of the first tenant. The tenants may include a third sub-tenant and a fourth sub-tenant where the sub-tenants are a part of the second tenant. At least two of the tenants may not be part of the same organization or company.

**[0029]** In some implementations, the instructions, when executed, further cause the processor to display, via the computing application, a graphical user interface associated with the computer where the graphical user interface displays the tenants in a single workspace, and wherein the tenants are displayed with a hierarchical structure. The graphical user interface may display each of the tenants tenant as an object where the object include an graphical input to add a sub-tenant to the respective tenants. The graphical user interface may display each of the tenants as an object where the object include an graphical input to add a sub-tenant to the respective tenants.

**[0030]** In some implementations, the instructions, when executed, further cause the processor to receive, in the workspace at the graphical user interface of the computing application, a first permission command to establish the permissions for a first tenant and a second permission command to establish the permissions for a second tenant. The graphical user interface may be configured to receive at least one permission setting selected from group consisting of a read permission, a write permission, and a modify permission.

**[0031]** In some implementations, the instructions, when executed, further cause the processor to cause, by a processor of the computing application, the permissions for the tenants to be set using the received first permission command and the received second permission command.

**[0032]** In some implementations, the instructions, when executed, further cause the processor to receive, in the workspace at the graphical user interface of the computing application, a third permission command to establish the permissions for a third tenant where the graphical user interface shows the first object and the second object as hierarchical subunits of the third object.

#### BRIEF DESCRIPTION OF THE FIGURES

**[0033]** The foregoing and other objects, aspects, features, and advantages of the present disclosure will become more apparent and better understood by referring to the following description taken in conjunction with the accompanying drawings, in which:

**[0034]** FIG. 1 is an example graphical user interface for establishing permissions using an organization matrices in accordance with an embodiment of the invention.

**[0035]** FIG. 2 is an example graphical user interface for establishing permissions using an organization matrices in accordance with another embodiment of the invention.

**[0036]** FIG. 3 is an example graphical user interface for establishing granular permission-controls for a given asset for an organization in accordance with an embodiment of the invention.

**[0037]** FIG. 4 is an example graphical user interface for establishing permissions controls for a class of asset for an organization in accordance with an embodiment of the invention.

**[0038]** FIG. 5 is an example graphical user interface for establishing permissions controls for a class of asset for an organization in accordance with another embodiment of the invention.

**[0039]** FIG. 6 is an example graphical user interface for establishing permissions controls for a class of asset for an organization in accordance with another embodiment of the invention.

**[0040]** FIG. 7 is an example graphical user interface for adding a given asset in accordance with another embodiment of the invention.

**[0041]** FIG. 8 is a flowchart of an example method of establishing permissions using an organization matrices in accordance with an embodiment of the invention.

**[0042]** FIG. 9 shows an example of a computing device and a mobile computing device that can be used to implement the techniques described in this disclosure.

**[0043]** The features and advantages of the present disclosure will become more apparent from the detailed description set forth below when taken in conjunction with the drawings, in which like reference characters identify corresponding elements throughout. In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements.

#### DETAILED DESCRIPTION

**[0044]** FIG. 1 is an example graphical user interface **100** for establishing permissions using an organization matrices in accordance with an embodiment of the invention.

**[0045]** The graphical user interface **100** includes a main workspace **102** that allows an end-user to establish permissions for a multi-tenant database associated to a given asset (or asset class) for one or more tenants and/or sub-tenants using a hierarchical structure **104**. The graphical user interface **100** may display the visibility permissions on a single workspace associated to a given device (or asset class). This display provides additional organization relationship not readily apparent in a list. To this end, an administrator setting the permission can readily identify participants that are incorrectly assigned permission to access data of the given device (or asset class).

**[0046]** In some implementations, an asset is an electronic device to which a given data is associated. For example, in some implementations, the data may be associated to industrial equipment, retail equipment vending machines, transport vehicles, personal computing devices, cellphones, tablets, power generation equipment, power distribution equipment, hospital equipment, medical equipment, computing equipment, rental car fleets, mining equipment, warehouse equipment, among others. As show, the asset is a type of mining machinery, referred to as "P1\_L2\_KETTLE" **105**.

**[0047]** In some implementations, a hierarchical structure **104** may include an organization chart. In some implementations, a hierarchical structure **104** may include Venn diagrams, three-dimensional charts, linked diagrams, tab-indented lists, and other spatial representations by which hierarchical relationships can be visually determined.

**[0048]** In some implementations, the graphical user interface **100** displays each of the tenants **106** and the sub-tenants **108** in an hierarchical diagram **104** (such as an organization chart). In some implementations, an hierarchical diagram **104** is a diagram that shows the structure of other sub-organizations in a given organization and relationships among them. In some implementations, a hierarchical diagram **104** is a

diagram that shows relationships among organizations as it defines whether elements of the organizations are shared or not shared. In some implementations, a hierarchical diagram 104 is made up of an organizational matrices of organization units.

[0049] A hierarchical diagram 104 may include one or more sub-units, which may have sub-units of its own. In some implementations, the primary unit in the main workspace 102 is the tenant 106, and the subunits are sub-tenants 108.

[0050] Each of the tenants 106 and the sub-tenants 108 may represent and allow permission controls for a given organization having a collection of members. A member may be represented as an individual user account in the system. The individual user account may be associated, for example, with or belong to a particular organization, company, corporation, government, entity, group, circle, user-base, or tag.

[0051] In some implementations, the user-accounts in the organization may belong to or be associated with a smaller distinct group within the organization and are referred to as a sub-organization. Examples of sub-organizations may be departments, teams, task forces, individuals, or partner organizations. Sub-organizations are considered to belong to the organization of which they are a part of. In some implementations, the organization may be referred to as the parent and the sub-organization is referred to as a child.

[0052] In some implementations, the hierarchical diagram 104 indicates the root parent organization in the hierarchy by placing it at the top of the chart. Children of the parent organization are indicated by a line drawn from the parent to the child graphically positioned below the parent. A given sub-organizations may include another sub-organization.

[0053] For example, as shown in this example, the hierarchical diagram 104 includes a tenant (or parent) “Peabody Company”, shown as “Peabody 110”, and sub-tenants (or child) “Engineering Division” and “Production Division”, shown as “Engineering” 112 and “Production” 114. The Engineering sub-organization 112 has a child “Heavy Equipment Sub-Divisions”, shown as “Heavy equipment” 116. The Production sub-organization 114 has children “South East Production” and “North East Production”, shown as “SE Production” 118 and “NE Production” 120. The SE Production sub-organization 122 has a child “Tampa Office,” shown as “Tampa” 122.

[0054] Although depicted as a top-down hierarchy in this implementation, the hierarchy diagram 104 may take any form which facilitates the modification of the organizational structure according to the hierarchical nature described. Thus, the hierarchy diagram 104 may be represented in a horizontal manner with the root parent positioned graphically left or right of the children. The hierarchy diagram 104 may be represented with the root parent positioned graphically below the children. Furthermore, the present disclosure contemplates other organizational charts which adequately represent the hierarchical relationships of organizations and sub-organizations within the organizational units.

[0055] In some implementations, the graphical user interface 100 provides an input 124a to add children (i.e., sub-tenant). A selection of the button 124a corresponding to the Peabody organization may, for example, create a child organization that is parallel in the hierarchy to Engineering and Production.

[0056] In some implementations, a tenant 106 or sub-tenant 108 may be “dragged and dropped” into the workspace from a list of existing organizations and sub-organizations.

[0057] To allow for collaborative data sharing among organizations, in some implementations, the interface may allow for co-tenants 126 to be added and displayed in the main workspace 102. To this end, permissions for organizations that are not associated to one another, rather than by a given asset, may be established for a single workspace. To this end, any inadvertent assignment of permission to an entity may be quickly identified by an end-user using the graphical user interface 100 to set permissions.

[0058] In some implementations, the graphical user interface 100 may update a sub-tenant workspace 128 in response to a selection of a tenant 106 or a sub-tenant 108 in the main workspace 102. The sub-tenant workspace 128 may identify the selected organization 132 in the main workspace 102 and a list of sub-tenants 134 associated to the selected item 132. In this example, an end-user has selected “Peabody” 110 in the workspace 102. To this end, the sub-tenant workspace 128 displays the two sub-tenants 112, 118, namely “Engineering” 112 and “Production” 114. The sub-tenant workspace 128 may include an input 136 to remove a given sub-tenant 108 from a given parent (i.e., the parent item 132 being displayed sub-organization workspace 128).

[0059] In some implementations, the graphical user interface 100 may include a tenant workspace to remove tenants, such as tenants 138 or 140 (corresponding to tenants 110 and 128), from the main workspace 102. As shown, the two tenants includes “Peabody” 110 and “Joy Mining” 106. Each of the tenants 106, 128 may include an input 142 for an end-user to remove a given tenant 106 from the main workspace 102.

[0060] In some implementations, the tenant workspace 136 may include an “add tenant” input 144. Upon selection of the input 144, the graphical user interface 100 may open a dialogue box for a user to select an organization to include from a list. Alternatively, the graphical user interface 100 may add a tenant icon (corresponding to a tenant 106) to the main workspace 102. The end-user can then select the added tenant icon (not shown) to modify its properties (including the corresponding organization) using the sub-tenant workspace 128 and/or the main workspace 102.

[0061] As shown, two co-tenant organizations 104, 126 are provided, including “Peabody” 106 and Joy Mining Company 126 (referred to as “Joy Mining” 128). Both “Peabody” 106 and “Joy Mining” 126 are root parent organizations. As co-tenants, the hierarchical diagram 104, 126 are not joined (e.g., shown in some implementations as a line) by an organizational link. “Joy Mining” 128 includes a “Service Division” sub-organization 1308 (referred to as “Service” 130).

[0062] In the main workspace 102, the end-user may add organizations (or sub-organizations) associated to either “Peabody” 110 or “Joy Mining” 128. To this end, an end-user may add sub-organizations to “Peabody” 106 or “Joy Mining” 126 using the “add” organization input 124. In some implementations, the selection of the input 124 may open a dialogue box to select an organization from an existing list or to add a new organization.

[0063] In some implementations, “Peabody” 110 and “Joy Mining” 128 may be depicted in the workspace 102 parallel to one another in the hierarchy. In some implementations, the graphical user interface 100 may display the hierarchical diagram 104, 106 only as a partial representation of the entire organizational chart. For example, in some implementations, only the tenants are shown. In another example, in some implementations, only the tenants 106 and a defined number

of sub-tenants **108**, specified with the graphical user interface **100** or by the end-user, are shown.

**[0064]** In some implementations, an organization may wish to grant permissions to particular sub-organizations or other organizations.

**[0065]** This may occur, for example, between companies engaged in an equipment service agreement. An example is a mining Company A that purchased mining equipment from a mining equipment Manufacturer B. Mining company A tracks its maintenance and operational data in the database. If mining Company A grants access to view the maintenance and operational data to manufacturer B, then Manufacturer B can use this information to improve its customer-service operations to mining company A. Manufacturer B may be able to use the maintenance and operational data to create or improve its best practices instructions regarding equipment usage, or to improve components which it observes are failing at a greater than expected rate. For instance, Manufacturer B may observe, directly through the maintenance and operational data collected in mining Company A's database, that bearings in its conveyor system are tending toward failure when operated for more than six hours continuously at ambient temperatures greater than 100° F. Manufacturer B can provide guidance to mining Company A that it should stagger operations at high ambient temperatures in order to allow the bearings to cool, or may determine that a modified lubrication regimen is required at high ambient temperatures, or may modify the construction of the equipment in future products. Both Manufacturer B and mining Company A benefit from this data sharing, since Manufacturer B is able to provide a higher quality of service, and mining company A reduces costs of equipment failure. Such data sharing arrangements may arise organically, or may be negotiated into contract. For example, Manufacturer B may negotiate such data collaborative endeavors into a purchase contract in exchange for some concession such as enhanced warranty service.

**[0066]** In another example, rather than mining Company A being the administrator, equipment Manufacturer B may be the administrator. Manufacturer B can make agreements with several mining companies (for example, Companies A, C, and D) to aggregate the maintenance and operational data to share among the companies, for example, as part of an after-market maintenance service agreement. Manufacturer B may provide a more efficient service in having shared equipment (i.e., less equipment to operate and maintain) to which the mining companies may benefit. Additionally, Manufacturer B may have determine inter-companies best practices or observe inter-companies issues in having aggregated data from all of the mining companies to which Manufacturer B can share in an aggregated manner with the mining companies. Additionally, manufacturer B may improve its products and services using the aggregated set of operational data.

**[0067]** Another example of an organizational relationship which would benefit from collaborative information sharing are naturally symbiotic industries, such as those in which there are producers of a product or equipment, and organizations which operate to service that producer's product. Examples of such industries are automobile manufacturers and automobile dealers, computing device manufacturers and computer service contractors, and the like. For example, an automobile dealer may provide information about maintenance operations and equipment failures, which the automobile manufacturer can use to suggest improved maintenance guidelines and to modify future equipment design. Further-

more, such information, when aggregated from multiple dealers, may enable the manufacturer to anticipate demand for components that are being frequently replaced due to failure, and increase production of replacement components.

**[0068]** An additional organizational structure which may benefit from information sharing are the various sub-organizations under a parent organization. For example, a food product factory may have a broad array of subsidiary companies or departments which all belong to the parent company. Such sub-organizations or departments may include human resources, management, maintenance, sanitation, equipment operators, medical personnel, quality control, shipping and receiving, sales, information technology, and engineering.

**[0069]** Another example of collaborative information sharing may be between a city government and the utility companies. An electricity company may, with consent of the customer, enable the government, universities, and/or government agencies to access billing and electricity utilization information. Such information may be aggregated and used, for example, to anticipate electricity outages, potential capacity problems, or the need for government subsidies in order to increase the level of service.

**[0070]** FIG. 2 is an example graphical user interface **100** for establishing permissions using an organization matrices in accordance with another embodiment of the invention.

**[0071]** In some implementations, the graphical user interface **100** may include an organization workspace **202** to edit properties and members of a given organization. The organization workspace **202** may display the hierarchical diagram **104** of a given organization, including the tenant and its sub-tenants.

**[0072]** When selected, a given tenant or sub-tenant may be highlighted or enunciated in the organization workspace **202**. In some implementations, the graphical user interface **100** may include a properties workspace **206** to display information relating to a given tenants **106** or sub-tenants **108** selected in the organization workspace **202**. The properties workspace **206** may allow the end user to rename (see, for example, via input **207**) the organization as being displayed. The end user may also as a description of the organization using the properties workspace **206**.

**[0073]** The main workspace **102** may include an members workspace **208** to display members of the organization. A member may be a part of a sub-tenant or a tenant and generally refers to actual end-users. The members workspace **208** may include an input **210** to add additional members. The members workspace **208** may display a list **212** of existing members within a selected organization. When there are no members included, the members workspace **208** may indicate that the selected organization includes no members having permissions.

**[0074]** In some implementations, the graphical user interface **100** may include setting granular permission controls **204** for a given tenant/sub-tenant. The granular permission controls **204** may include read access, write access, modify access, and change permissions access for an organization or its members and may be specific for operation during design time and during run-time.

**[0075]** In some implementations, if visibility permission is granted to an organization unit, then visibility permission/access is also extended to members of the organizations. The granular security-model may then be employed to grant specific permissions to any users/members of that organization unit.

[0076] FIG. 3 is an example graphical user interface 100 for establishing granular permissions controls for a given asset for an organization in accordance with an embodiment of the invention. Specifically, the graphical user interface 100 includes a workspace 300 to display a list of organizations and/or tenants and sub-tenants to which visibility permission is provided. The workspace 300 may include an input 302 to remove a given organization 304 from the list. The workspace 300 may include an input 314 to add permissions for an organization or organization unit. The input 314 may open the main workspace 102, as described in relation to FIG. 1.

[0077] The graphical user interface 100 may include a workspace to display a list of organizations and/or tenants to which design-time permission or run-time permission is provided. The design-time permission may be displayed when an end-user selects “design-time” input 310 from the granular permission controls 204. The run-time permission may be displayed when an end-user selects “run-time” input 312 from the granular permission controls 204.

[0078] In another aspect of an embodiment of the invention, the graphical user interface 100 may allow a user to configure a class of assets.

[0079] FIG. 4 is an example graphical user interface 100 for establishing permissions controls for a class of asset for an organization in accordance with an embodiment of the invention.

[0080] In some implementations, a class of assets may refer to a type, rather than individual instance, of a given electronic device. For example, where an asset may be a specific object (for example, a truck), a class of assets may correspond to multiple objects having a set of common properties defined by the class/template. The class of assets may be defined by a data template of the object in which the data template includes one or more properties that define the class/template. For example, The template may include common properties and information that determines or defines what constitute a truck.

[0081] In some implementations, the graphical user interface 100 includes a workspace 402 for managing classes of assets. The classes of assets may be part of a development software to build and manage a data-and information-model of assets and classes. In some implementations, the class of assets is referred to as a “Thing”. A “Thing” may be defined as an instance of a “Thing Template.” A “Thing Template” may be an abstract class that can inherit from one or more “Thing Shapes,” which is defined by a set of properties, services, and events, “Thing Template,” and “Thing instance.” To this end, if a “Thing Template” inherits from one or more “Thing Shapes”, all of the properties, events, and services of the “Thing Shapes” are part of the “Thing Template.” When a “Thing instance” is created from a “Thing Template”, all of the properties, events, and services of the “Thing Template” are realized within the “Thing instance.” Thus, if a new service, property, or capability is defined at the “Thing Shape” or “Thing Template” level, each “Thing” instance that is derived from those entities immediately inherits that service, property or capability. To this end, once a new “Thing” is defined in the model, the full set of services and data for the “Thing” is available as a class. Examples of methods of class of assets, as “Things”, are described in co-pending and concurrently filed U.S. patent application, titled “SYSTEM AND METHOD OF ABSTRACTING COMMUNICATION PROTOCOL USING SELF-DESCRIBING MESSAGES”, filed Mar. 21, 2014, naming

inventors Rick Bullotta, John Canosa, Bob DeRemer, and Mike Mahoney, and having attorney docket no. 2009132-0037. The text of this application is incorporated by reference in their entirety.

[0082] In some implementations, a class of assets may include a set of machinery at an industrial complex having data stored in a given database; a set of computer or an office equipment at a business or government office having data stored in a given database; a set of point-of-sale machine at a market place or vending machines having data stored in a given database; a set of construction equipment or vehicles having data stored in a given database; a set of power generation or distribution equipment having data stored in a given database; a set of power substation or a transmission equipment having data stored in a given database; a set of building meter having data stored in a given database; a set of server having operational data stored in a given database; a set of networking or routing equipment having data stored in a given database; a set of smart appliance having data stored in a given database; a set of exercise machines having data stored in a given database; a set of medical device or prosthesis devices having data stored in a given database; a set of medical diagnostic devices or hospital equipment having data stored in a given database; a set of commercial vehicles or transport containers having data stored in a given database; a set of motor vehicles or electric bicycles having data stored in a given database; a set of cellphones having data stored in a given database, a set of laptops having data stored in a given database, a set of tablet having data stored in a given database, a set of electronic readers having data stored in a given database; or a set of clothing electronic-tag having data stored in a given database.

[0083] Referring back to FIG. 4, the workspace 404 includes example types of assets and classes of assets. These assets and classes of assets include sensors 408, security equipment 410, vending machines 412, medical equipment 414, vehicles 416, plant equipment 418, mining equipment 420, and factory machinery 422. A sensor equipment 408, for example, is shown as “AB Test Remote Tunneling Thing” 408. A security equipment 410 is shown as “Security Monitor” 410. A vending machine 412 is shown as “VM-007 Generic Thing” 412. A medical equipment 414 is shown as “CT-2” 414. A class of vehicles 416 is shown as “Vehicle 42” 416. A plant equipment 418 is shown as “P1 L2 Kettle” 418 (referring to a kettle machine on line 2 of a given plant 1). A mining equipment 420 is shown as “Station 3 LOCI” 420. A factory equipment 422 is shown as “Plant 5” 422 (referring to a factory in Yengchang, China).

[0084] In some implementations, the graphical user interface 100 may include a search function 424 to search among the classes of assets within the multi-tenant database.

[0085] In some implementations, the graphical user interface 100 may include tabs to navigate among the workspaces of the graphical user interface 100.

[0086] In some implementations, the graphical user interface 100 may include permission controls 406, such as for design-time permissions, run-time permissions, and visibility permissions, as described in relation to a given asset in FIG. 3.

[0087] FIG. 5 is an example graphical user interface 100 for establishing permissions controls for a class of asset for an organization in accordance with another embodiment of the invention.



[0088] In some implementations, the graphical user interface 100 includes a workspace 502 to configure permissions for a class of members for a given asset or a class of assets. As shown, the workspace 502 displays visibility permissions for a class of members, including “Everyone” 506 and “Development” 508 for a general class of assets. Such class of assets may be a “Things” 510, as described in relation to FIG. 4.

[0089] FIG. 6 is an example graphical user interface 100 for establishing permissions controls for a class of asset for an organization in accordance with another embodiment of the invention.

[0090] In some implementations, the graphical user interface 100 includes a permission panel 602 to establish permissions for a class of assets. In some implementations, the permission panel 602 includes a list of visibility permission given to a set of organization units.

[0091] As shown, the graphical user interface 100 is configured to set permissions for a class of factory equipment “Kettle” 604. The permission is provided to organization 606, such as the “Tampa office” 122 of the Peabody South East Production Division,” as described in FIG. 1. The “Tampa office” 122 is shown as “Peabody:Tampa” 122. The workspace 602 includes a remove button 608 to remove Peabody: Tampa from the list of organizational units with visibility permission.

[0092] In some implementations, the graphical user interface 100 includes an add button 610 (e.g., “Add Org/Org Units” 610) to add members to the list to visibility permissions given for a class of assets. In some implementations, the selection may be made using “drag and drop” mechanism, checkbox selection, and text entry types of interface.

[0093] FIG. 7 is an example graphical user interface 100 for adding a given asset in accordance with another embodiment of the invention.

[0094] In some implementations, the graphical user interface 100 includes a workspace 702 to add a given asset derived from an asset class. The workspace 702 may include a name field 704, a description field 706, a tag field 708, an asset template 710 (e.g., a “Thing Template”), and a Datashape definition 712.

[0095] In some implementations, the name field 704 provides a name of a given asset. As shown, the asset is an “P1\_L2\_Kettle,” as described in relation to FIG. 1.

[0096] In some implementations, the tag field 708 provides metadata labels for the asset (or class of assets). For example, as shown, the “P1\_L2\_Kettle” 105 includes a type of equipment tag 714 as well as a location tag 716. The equipment tag 714, here, defines “P1\_L2\_Kettle” as a type of kettle manufacturing equipment (“KettleMfg” located at “Walker EP”). The location tag 716, here, defines “P1\_L2\_Kettle” as a type of application (located at “Acme Manufacturing”).

[0097] In some implementations, the asset template 710 provides a definition of the asset class (here, shown as “Kettle”), as described in relation to FIG. 6. The asset template 710 may be a ThingTemplate, as described in relation to FIG. 4.

[0098] In some implementations, the Datashape definition 712 provides a definition of a DataShape, as described in relation to FIG. 4.

[0099] FIG. 8 is a flowchart of an example method 800 of establishing permissions using an organization matrices in accordance with an embodiment of the invention.

[0100] In some implementations, the method 800 includes providing, by a processor of a computer, a computing appli-

cation for setting permissions for accessing data entries of a multi-tenant database (step 802). The data entries may store data associated with a number of computing devices where the multi-tenant database is accessible by two or more tenants (such as 104), as described in relation to FIG. 1. The two or more tenants may include a first sub-tenant and a second sub-tenant where the first sub-tenant and the second sub-tenant are a part of the first tenant. The tenants may include a third sub-tenant and a fourth sub-tenant where the third sub-tenant and the fourth sub-tenant are a part of the second tenant.

[0101] In some implementations, at least two of the tenants may not be a part of the same organization. The graphical user interface 100 may be configured to receive at least one permission setting selected from group consisting of a read permission, a write permission, and a modify permission.

[0102] In some implementations, the method 800 includes displaying, via the computing application, a graphical user interface 100 associated with the computer where the graphical user interface 100 displays the one or more tenants (such as 104) in a single workspace (such as 102) and where the tenants are displayed with a hierarchical structure (step 804). The graphical user interface 100 may display each of the first tenant and second tenant as an object where the object include an graphical input to add a sub-tenant to the respective first tenant and second tenant.

[0103] In some implementations, the method 800 includes receiving, in the workspace at the graphical user interface, a first permission command to establish the permissions for a first tenant and a second permission command to establish the permissions for a second tenant of the two or more tenants (step 806).

[0104] In some implementations, the method 800 includes causing, by a processor of the computing application, the permissions for the first tenant and the second tenant to be set using the received first permission command and the received second permission command (step 808).

[0105] In some implementations, the method may further include receiving, in the workspace at the graphical user interface of the computing application, a third permission command to establish the permissions for a third tenant where the graphical user interface shows the first object and the second object as hierarchical subunits of the third object.

[0106] FIG. 9 shows an example of a computing device 900 and a mobile computing device 950 that can be used to implement the techniques described in this disclosure. The computing device 900 is intended to represent various forms of digital computers, such as laptops, desktops, workstations, personal digital assistants, servers, blade servers, mainframes, and other appropriate computers. The mobile computing device 950 is intended to represent various forms of mobile devices, such as personal digital assistants, cellular telephones, smart-phones, and other similar computing devices. The components shown here, their connections and relationships, and their functions, are meant to be examples only, and are not meant to be limiting.

[0107] The computing device 900 includes a processor 902, a memory 904, a storage device 906, a high-speed interface 908 connecting to the memory 904 and multiple high-speed expansion ports 910, and a low-speed interface 912 connecting to a low-speed expansion port 914 and the storage device 906. Each of the processor 902, the memory 904, the storage device 906, the high-speed interface 909, the high-speed expansion ports 910, and the low-speed interface 912, are

interconnected using various busses, and may be mounted on a common motherboard or in other manners as appropriate. The processor 902 can process instructions for execution within the computing device 900, including instructions stored in the memory 904 or on the storage device 906 to display graphical information for a GUI on an external input/output device, such as a display 916 coupled to the high-speed interface 908. In other implementations, multiple processors and/or multiple buses may be used, as appropriate, along with multiple memories and types of memory. Also, multiple computing devices may be connected, with each device providing portions of the necessary operations (e.g., as a server bank, a group of blade servers, or a multi-processor system).

**[0108]** The memory 904 stores information within the computing device 900. In some implementations, the memory 904 is a volatile memory unit or units. In some implementations, the memory 904 is a non-volatile memory unit or units. The memory 904 may also be another form of computer-readable medium, such as a magnetic or optical disk.

**[0109]** The storage device 906 is capable of providing mass storage for the computing device 900. In some implementations, the storage device 906 may be or contain a computer-readable medium, such as a floppy disk device, a hard disk device, an optical disk device, or a tape device, a flash memory or other similar solid state memory device, or an array of devices, including devices in a storage area network or other configurations. Instructions can be stored in an information carrier. The instructions, when executed by one or more processing devices (for example, processor 902), perform one or more methods, such as those described above. The instructions can also be stored by one or more storage devices such as computer- or machine-readable mediums (for example, the memory 904, the storage device 906, or memory on the processor 902).

**[0110]** The high-speed interface 909 manages bandwidth-intensive operations for the computing device 900, while the low-speed interface 912 manages lower bandwidth-intensive operations. Such allocation of functions is an example only. In some implementations, the high-speed interface 909 is coupled to the memory 904, the display 916 (e.g., through a graphics processor or accelerator), and to the high-speed expansion ports 910, which may accept various expansion cards (not shown). In the implementation, the low-speed interface 912 is coupled to the storage device 906 and the low-speed expansion port 914. The low-speed expansion port 914, which may include various communication ports (e.g., USB, Bluetooth®, Ethernet, wireless Ethernet) may be coupled to one or more input/output devices, such as a keyboard, a pointing device, a scanner, or a networking device such as a switch or router, e.g., through a network adapter.

**[0111]** The computing device 900 may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a standard server 920, or multiple times in a group of such servers. In addition, it may be implemented in a personal computer such as a laptop computer 922. It may also be implemented as part of a rack server system 924. Alternatively, components from the computing device 900 may be combined with other components in a mobile device (not shown), such as a mobile computing device 950. Each of such devices may contain one or more of the computing device 900 and the mobile computing device 950, and an entire system may be made up of multiple computing devices communicating with each other.

**[0112]** The mobile computing device 950 includes a processor 952, a memory 964, an input/output device such as a display 954, a communication interface 966, and a transceiver 968, among other components. The mobile computing device 950 may also be provided with a storage device, such as a micro-drive or other device, to provide additional storage. Each of the processor 952, the memory 964, the display 954, the communication interface 966, and the transceiver 968, are interconnected using various buses, and several of the components may be mounted on a common motherboard or in other manners as appropriate.

**[0113]** The processor 952 can execute instructions within the mobile computing device 950, including instructions stored in the memory 964. The processor 952 may be implemented as a chipset of chips that include separate and multiple analog and digital processors. The processor 952 may provide, for example, for coordination of the other components of the mobile computing device 950, such as control of user interfaces, applications run by the mobile computing device 950, and wireless communication by the mobile computing device 950.

**[0114]** The processor 952 may communicate with a user through a control interface 958 and a display interface 956 coupled to the display 954. The display 954 may be, for example, a TFT (Thin-Film-Transistor Liquid Crystal Display) display or an OLED (Organic Light Emitting Diode) display, or other appropriate display technology. The display interface 956 may comprise appropriate circuitry for driving the display 954 to present graphical and other information to a user. The control interface 958 may receive commands from a user and convert them for submission to the processor 952. In addition, an external interface 962 may provide communication with the processor 952, so as to enable near area communication of the mobile computing device 950 with other devices. The external interface 962 may provide, for example, for wired communication in some implementations, or for wireless communication in other implementations, and multiple interfaces may also be used.

**[0115]** The memory 964 stores information within the mobile computing device 950. The memory 964 can be implemented as one or more of a computer-readable medium or media, a volatile memory unit or units, or a non-volatile memory unit or units. An expansion memory 974 may also be provided and connected to the mobile computing device 950 through an expansion interface 972, which may include, for example, a SIMM (Single In Line Memory Module) card interface. The expansion memory 974 may provide extra storage space for the mobile computing device 950, or may also store applications or other information for the mobile computing device 950. Specifically, the expansion memory 974 may include instructions to carry out or supplement the processes described above, and may include secure information also. Thus, for example, the expansion memory 974 may be provide as a security module for the mobile computing device 950, and may be programmed with instructions that permit secure use of the mobile computing device 950. In addition, secure applications may be provided via the SIMM cards, along with additional information, such as placing identifying information on the SIMM card in a non-hackable manner.

**[0116]** The memory may include, for example, flash memory and/or NVRAM memory (non-volatile random access memory), as discussed below. In some implementations, instructions are stored in an information carrier. that the

instructions, when executed by one or more processing devices (for example, processor 952), perform one or more methods, such as those described above. The instructions can also be stored by one or more storage devices, such as one or more computer- or machine-readable mediums (for example, the memory 964, the expansion memory 974, or memory on the processor 952). In some implementations, the instructions can be received in a propagated signal, for example, over the transceiver 968 or the external interface 962.

[0117] The mobile computing device 950 may communicate wirelessly through the communication interface 966, which may include digital signal processing circuitry where necessary. The communication interface 966 may provide for communications under various modes or protocols, such as GSM voice calls (Global System for Mobile communications), SMS (Short Message Service), EMS (Enhanced Messaging Service), or MMS messaging (Multimedia Messaging Service), CDMA (code division multiple access), TDMA (time division multiple access), PDC (Personal Digital Cellular), WCDMA (Wideband Code Division Multiple Access), CDMA2000, or GPRS (General Packet Radio Service), among others. Such communication may occur, for example, through the transceiver 968 using a radio-frequency. In addition, short-range communication may occur, such as using a Bluetooth®, Wi-Fi™, or other such transceiver (not shown). In addition, a GPS (Global Positioning System) receiver module 970 may provide additional navigation- and location-related wireless data to the mobile computing device 950, which may be used as appropriate by applications running on the mobile computing device 950.

[0118] The mobile computing device 950 may also communicate audibly using an audio codec 960, which may receive spoken information from a user and convert it to usable digital information. The audio codec 960 may likewise generate audible sound for a user, such as through a speaker, e.g., in a handset of the mobile computing device 950. Such sound may include sound from voice telephone calls, may include recorded sound (e.g., voice messages, music files, etc.) and may also include sound generated by applications operating on the mobile computing device 950.

[0119] The mobile computing device 950 may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a cellular telephone 980. It may also be implemented as part of a smart-phone 982, personal digital assistant, or other similar mobile device.

[0120] Various implementations of the systems and techniques described here can be realized in digital electronic circuitry, integrated circuitry, specially designed ASICs (application specific integrated circuits), computer hardware, firmware, software, and/or combinations thereof. These various implementations can include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device.

[0121] These computer programs (also known as programs, software, software applications or code) include machine instructions for a programmable processor, and can be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the terms machine-readable medium and computer-readable medium refer to any com-

puter program product, apparatus and/or device (e.g., magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The term machine-readable signal refers to any signal used to provide machine instructions and/or data to a programmable processor.

[0122] To provide for interaction with a user, the systems and techniques described here can be implemented on a computer having a display device (e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor) for displaying information to the user and a keyboard and a pointing device (e.g., a mouse or a trackball) by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback (e.g., visual feedback, auditory feedback, or tactile feedback); and input from the user can be received in any form, including acoustic, speech, or tactile input.

[0123] The systems and techniques described here can be implemented in a computing system that includes a back end component (e.g., as a data server), or that includes a middleware component (e.g., an application server), or that includes a front end component (e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the systems and techniques described here), or any combination of such back end, middleware, or front end components. The components of the system can be interconnected by any form or medium of digital data communication (e.g., a communication network). Examples of communication networks include a local area network (LAN), a wide area network (WAN), and the Internet.

[0124] The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[0125] In view of the structure, functions and apparatus of the systems and methods described here, in some implementations, a system and method for determining transaction approval based in part upon consumer purchase behavior are provided. Having described certain implementations of methods and apparatus for supporting transaction approval determination, it will now become apparent to one of skill in the art that other implementations incorporating the concepts of the disclosure may be used. Therefore, the disclosure should not be limited to certain implementations, but rather should be limited only by the spirit and scope of the following claims.

What is claimed:

1. A computer-implemented method comprising:
  - providing, by a processor of a computing device, a graphical user interface for setting permissions for accessing data entries of a multi-tenant database,
  - the data entries storing data associated with a plurality of computing devices, wherein
  - the multi-tenant database is accessible by two or more tenants;
  - displaying, by the processor, the two or more tenants in a single workspace, wherein the two or more tenants are displayed in a hierarchical structure;

- receiving, by the processor, via the single workspace, a first permission command associated to a first tenant of the two or more tenants and a second permission command associated to a second tenant of the two or more tenants; and
- causing, by the processor, the permissions for the first tenant and the second tenant to be set using the received first permission command and the received second permission command.
2. The computer-implemented method of claim 1, wherein the two or more tenants includes a first sub-tenant and a second sub-tenant, wherein the first sub-tenant and the second sub-tenant are a part of the first tenant.
3. The computer-implemented method of claim 1, wherein the two or more tenants includes a third sub-tenant and a fourth sub-tenant, wherein the third sub-tenant and the fourth sub-tenant are a part of the second tenant.
4. The computer-implemented method of claim 1, wherein at least two of the two or more tenants are not part of the same organization.
5. The computer-implemented method of claim 1 further comprising:
- receiving, via the workspace, a third permission command associated to a third tenant of the two or more tenants, wherein
- the graphical user interface shows the first object and the second object as hierarchical subunits of the third object.
6. The computer-implemented method of claim 1, wherein the graphical user interface is configured to receive at least one permission setting selected from group consisting of a read permission, a write permission, and a modify permission.
7. The computer-implemented method of claim 1, wherein the graphical user interface displays each of the first tenant and second tenant as an object, wherein the object includes an graphical input to add a sub-tenant to the respective first tenant and second tenant.
8. A system comprising:
- a processor; and
- a memory, the memory storing instructions that, when executed by the processor, cause the processor to:
- provide a graphical user interface for setting permissions for accessing data entries of a multi-tenant database, the data entries storing data associated with a plurality of computing devices, wherein
- the multi-tenant database is accessible by two or more tenants;
- display the two or more tenants in a single workspace, wherein the two or more tenants are displayed in a hierarchical structure;
- receive, via the single workspace, a first permission command associated to a first tenant of the two or more tenants and a second permission command associated to a second tenant of the two or more tenants; and
- cause the permissions for the first tenant and the second tenant to be set using the received first permission command and the received second permission command.
9. The system of claim 8, wherein the two or more tenants include a first sub-tenant and a second sub-tenant, wherein the first sub-tenant and the second sub-tenant are a part of the first tenant.
10. The system of claim 8, wherein the two or more tenants include a third sub-tenant and a fourth sub-tenant, wherein the third sub-tenant and the fourth sub-tenant are a part of the second tenant.
11. The system of claim 8, wherein at least two of the two or more tenants are not part of the same organization.
12. The system of claim 8, wherein the memory further stores instructions to:
- receive, via the workspace of the graphical user interface, a third permission command associated to a third tenant of the two or more tenants, wherein
- the graphical user interface shows the first object and the second object as hierarchical subunits of the third object.
13. The system of claim 8, wherein the graphical user interface is configured to receive at least one permission setting selected from group consisting of a read permission, a write permission, and a modify permission.
14. The system claim 8, wherein graphical user interface displays each of the first tenant and second tenant as an object, wherein the object includes an graphical input to add a sub-tenant to the respective first tenant and second tenant.
15. A non-transitory computer readable medium having instructions stored thereon, wherein the instructions, when executed by a processor, cause the processor to:
- provide a computing application for setting permissions for accessing data entries of a multi-tenant database, the data entries storing data associated with a plurality of computing devices, wherein
- the multi-tenant database is accessible by two or more tenants;
- display a graphical user interface associated with the computer, wherein
- the graphical user interface displays the two or more tenants in a single workspace, and wherein the two or more tenants are displayed with a hierarchical structure;
- receive, in the workspace, a first permission command to establish the permissions for a first tenant of the two or more tenants and a second permission command to establish the permissions for a second tenant of the two or more tenants; and
- cause the permissions for the first tenant and the second tenant to be set using the received first permission command and the received second permission command.
16. The computer readable medium of claim 15, wherein the two or more tenants include a first sub-tenant and a second sub-tenant, wherein the first sub-tenant and the second sub-tenant are a part of the first tenant.
17. The computer readable medium of claim 15, wherein the two or more tenants include a third sub-tenant and a fourth sub-tenant, wherein the third sub-tenant and the fourth sub-tenant are a part of the second tenant.
18. The computer readable medium of claim 15, wherein at least two of the two or more tenants are not part of the same organization.
19. The computer readable medium of claim 15, wherein the computer readable medium stores instructions thereon, wherein the instructions, when executed by a processor, cause the processor to:
- receive, in the workspace at the graphical user interface of the computing application, a third permission command to establish the permissions for a third tenant of the two or more tenants, wherein

the graphical user interface shows the first object and the second object as hierarchical subunits of the third object.

**20.** The computer readable medium of claim **15**, wherein the graphical user interface is configured to receive at least one permission setting selected from group consisting of a read permission, a write permission, and a modify permission.

**21.** The computer readable medium of claim **15**, wherein graphical user interface displays each of the first tenant and second tenant as an object, wherein the object includes an graphical input to add a sub-tenant to the respective first tenant and second tenant.

\* \* \* \* \*