

(12) STANDARD PATENT
(19) AUSTRALIAN PATENT OFFICE

(11) Application No. **AU 2012238127 B2**

(54) Title
Recovery of tenant data across tenant moves

(51) International Patent Classification(s)
G06F 17/40 (2006.01) **G06F 12/16** (2006.01)

(21) Application No: **2012238127** (22) Date of Filing: **2012.03.03**

(87) WIPO No: **WO12/134711**

(30) Priority Data

(31) Number (32) Date (33) Country
13/077,620 **2011.03.31** **US**

(43) Publication Date: **2012.10.04**

(44) Accepted Journal Date: **2017.02.02**

(71) Applicant(s)
Microsoft Technology Licensing, LLC

(72) Inventor(s)
Shah, Siddharth Rajendra;Da Silva Jr., Antonio Marco;Voronkov, Nikita;Taranov, Viktoriya;Blood, Daniel

(74) Agent / Attorney
Davies Collison Cave Pty Ltd, Level 15 1 Nicholson Street, MELBOURNE, VIC, 3000

(56) Related Art
US 7546354 B1



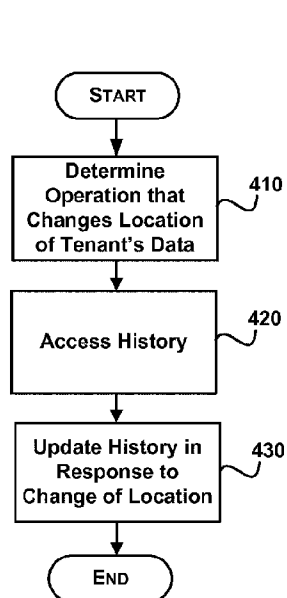
- (51) International Patent Classification:
G06F 17/40 (2006.01) *G06F 12/16* (2006.01)
- (21) International Application Number:
PCT/US2012/027637
- (22) International Filing Date:
3 March 2012 (03.03.2012)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
13/077,620 31 March 2011 (31.03.2011) US
- (71) Applicant (for all designated States except US): **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).
- (72) Inventors: **SHAH, Siddharth Rajendra**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **DA SILVA JR., Antonio Marco**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Red-

mond, Washington 98052-6399 (US). **VORONKOV, Nikita**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **TARANOV, Viktoriya**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **BLOOD, Daniel**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

[Continued on next page]

- (54) Title: RECOVERY OF TENANT DATA ACROSS TENANT MOVES



(57) Abstract: A history of locations of tenant data is maintained. The tenant data comprises data that is currently being used by the tenant and the corresponding backup data. When a tenant's data is changed from one location to another, a location and a time is stored within the history that may be accessed to determine a location of the tenant's data at a specified time. Different operations trigger a storing of a location/time within the history. Generally, an operation that changes a location of the tenant's data triggers the storing of the location within the history (e.g. upgrade of farm, move of tenant, adding a tenant, load balancing of the data, and the like). When tenant data is needed for an operation (e.g. restore), the history may be accessed to determine the location of the data.

FIG. 4



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

— as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

Published:

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

— before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

Declarations under Rule 4.17:

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

2012238127 28 Oct 2016

- 1 -

RECOVERY OF TENANT DATA ACROSS TENANT MOVES

BACKGROUND

[0001] Tenant data may be moved to different locations for various reasons. For example, tenant data may be moved when upgrading a farm, more space is needed for the tenant's data, and the like. In such cases, a new backup of the tenant data is made.

[0001a] It is desired to address or ameliorate one or more disadvantages or limitations associated with the prior art, or to at least provide a useful alternative.

10 SUMMARY

[0002] In accordance with the present invention there is provided a method executing on a processor for recovering tenant data across tenant moves, comprising:

performing full backup operations on the tenant data to generate tenant backup data according to a full backup schedule;

15 performing incremental backup operations on the tenant data to generate the tenant backup data according to an incremental backup schedule;

determining an operation that changes a storage location of where a tenant's data is stored within a network storage location to a different network storage location;

20 in response to the operation that changes the storage location of the tenant's data, updating a history for the tenant of the storage location of the tenant's data by adding a current storage location of where the tenant's data is currently stored, wherein the history identifies the current storage location and a time associated with when the tenant's data was stored at the current storage location and a backup location and a time associated with when the tenant's backup data was stored at the backup location;

25 when requested, accessing the history for the tenant to determine a previous storage location of where the tenant's data was stored based at least in part on a time indicating when the tenant's data was deleted; and

restoring the tenant's data using the previous storage location.

2012238127 28 Oct 2016

- 1A -

[0002a] The present invention also provides a computer-readable storage medium, excluding a signal, storing computer-executable instructions for recovering tenant data across tenant moves, comprising:

performing full backup operations on the tenant data to generate tenant
5 backup data according to a full backup schedule;

performing incremental backup operations on the tenant data to generate the
tenant backup data according to an incremental backup schedule;

determining an operation that changes a storage location of a tenant's data
from a network storage location to a different network storage location;

10 updating a history for the tenant to include a current storage location of the
tenant's data, wherein the history for the tenant includes a record for each storage location
at which the tenant's data has been stored and the current storage location, wherein each
record comprises a tenant data storage location, a backup storage location for the tenant
backup data, and time information indicating when the data was at each of the storage
15 locations;

when requested, accessing the history to determine a previous storage
location of the tenant's data using a time indicating when the tenant's data was deleted;
and

restoring the tenant's data using the previous storage location.

20 [0002b] The present invention also provides a system for recovering tenant data across
tenant moves, comprising:

a network connection that is configured to connect to a network;

a processor, memory, and a computer-readable storage medium;

25 an operating environment stored on the computer-readable storage medium
and executing on the processor;

a data store storing tenant data that is associated with different tenants; and

a backup manager operating that is configured to perform actions
comprising:

performing full backup operations on the tenant data to generate
30 tenant backup data according to a full backup schedule;

2012238127 28 Oct 2016

- 1B -

performing incremental backup operations on the tenant data to generate the tenant backup data according to an incremental backup schedule

receiving a request for a tenant's data within a network data store;

5 accessing a history of the tenant including tenant data storage locations to determine a storage location of the requested tenant data using a time indicating when the tenant's data was moved to backup storage, wherein the history includes a record for each storage location at which the tenant's data has been stored and the current storage location, wherein the record comprises a tenant data storage location, a backup storage location for the tenant backup data, and time
10 information indicating when the data was at each of the storage locations; and restoring the tenant's data using the storage location.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Some embodiments of the present invention are hereinafter described, by way of
15 non-limiting example only, with reference to the accompanying drawings, in which:

[0004] FIGURE 1 illustrates an exemplary computing environment;

[0005] FIGURE 2 shows a system for maintaining a location of tenant data across tenant moves;

[0006] FIGURE 3 shows a history including records for tenant data location changes;

20 [0007] FIGURE 4 illustrates a process for updating a history of a tenant's data location change; and

[0007a] FIGURE 5 shows a process for processing a request for restoring tenant data from a backup location.

25 DETAILED DESCRIPTION

[0008] A history of locations of tenant data is maintained. The tenant data comprises data that is currently being used by the tenant and the corresponding backup data. When a tenant's data is changed from one location to another, a location and a time is stored within the history that may be accessed to determine a location of the tenant's data at a specified
30 time. Different operations trigger a storing of a location/time within the history. Generally, an operation that changes a location of the tenant's data triggers the storing of the location

- 1C -

within the history (e.g. upgrade of farm, move of tenant, adding a tenant, load balancing of the data, and the like). When tenant data is needed for an operation (e.g. restore), the history may be accessed to determine the location of the data.

- [0009] Referring now to the drawings, in which like numerals represent like elements,
- 5 various embodiment will be described. In particular, FIGURE 1 and the corresponding discussion are intended to provide a brief, general description of a suitable computing environment in which embodiments may be implemented.

[0010] Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Other computer system configurations may also be used, including hand-held devices, multiprocessor systems, microprocessor-based or
5 programmable consumer electronics, minicomputers, mainframe computers, and the like. Distributed computing environments may also be used where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

10 [0011] Referring now to FIGURE 1, an illustrative computer environment for a computer 100 utilized in the various embodiments will be described. The computer environment shown in FIGURE 1 includes computing devices that each may be configured as a mobile computing device (e.g. phone, tablet, net book, laptop), server, a desktop, or some other type of computing device and includes a central processing unit
15 ("CPU"), a system memory 7, including a random access memory 9 ("RAM") and a read-only memory ("ROM") 10, and a system bus 12 that couples the memory to the central processing unit ("CPU") 5.

[0012] A basic input/output system containing the basic routines that help to transfer information between elements within the computer, such as during startup, is stored in the
20 ROM 10. The computer 100 further includes a mass storage device 14 for storing an operating system 16, application(s) 24, Web browser 25, and backup manager 26 which will be described in greater detail below.

[0013] The mass storage device 14 is connected to the CPU 5 through a mass storage controller (not shown) connected to the bus 12. The mass storage device 14 and its
25 associated computer-readable media provide non-volatile storage for the computer 100. Although the description of computer-readable media contained herein refers to a mass storage device, such as a hard disk or CD-ROM drive, the computer-readable media can be any available media that can be accessed by the computer 100.

[0014] By way of example, and not limitation, computer-readable media may
30 comprise computer storage media and communication media. Computer storage media includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, Erasable Programmable Read Only Memory

(“EPROM”), Electrically Erasable Programmable Read Only Memory (“EEPROM”), flash memory or other solid state memory technology, CD-ROM, digital versatile disks (“DVD”), or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer 100.

[0015] Computer 100 operates in a networked environment using logical connections to remote computers through a network 18, such as the Internet. The computer 100 may connect to the network 18 through a network interface unit 20 connected to the bus 12. The network connection may be wireless and/or wired. The network interface unit 20 may also be utilized to connect to other types of networks and remote computer systems. The computer 100 may also include an input/output controller 22 for receiving and processing input from a number of other devices, including a keyboard, mouse, or electronic stylus (not shown in FIGURE 1). Similarly, an input/output controller 22 may provide input/output to a display screen 23, a printer, or other type of output device.

[0016] As mentioned briefly above, a number of program modules and data files may be stored in the mass storage device 14 and RAM 9 of the computer 100, including an operating system 16 suitable for controlling the operation of a computer, such as the WINDOWS 7®, WINDOWS SERVER®, or WINDOWS PHONE 7® operating system from MICROSOFT CORPORATION of Redmond, Washington. The mass storage device 14 and RAM 9 may also store one or more program modules. In particular, the mass storage device 14 and the RAM 9 may store one or more application programs, including one or more application(s) 24 and Web browser 25. According to an embodiment, application 24 is an application that is configured to interact with an online service, such as a business point of solution service that provides services for different tenants. Other applications may also be used. For example, application 24 may be a client application that is configured to interact with data. The application may be configured to interact with many different types of data, including but not limited to: documents, spreadsheets, slides, notes, and the like.

[0017] Network store 27 is configured to store tenant data for tenants. Network store 27 is accessible to one or more computing devices/users through IP network 18. For example, network store 27 may store tenant data for one or more tenants for an online service, such as online service 17. Other network stores may also be configured to store data for tenants. Tenant data may also move from one network store to another network store

[0018] Backup manager 26 is configured to maintain locations of tenant data within a history, such as history 21. Backup manager 26 may be a part of an online service, such as online service 17, and all/some of the functionality provided by backup manager 26 may be located internally/externally from an application. The tenant data comprises data that is currently being used by the tenant and the corresponding backup data. When a tenant's data is changed from one location to another, a location and a time is stored within the history 21 that may be accessed to determine a location of the tenant's data at a specified time. Different operations trigger a storing of a location/time within the history. Generally, an operation that changes a location of the tenant's data triggers the storing of the location within the history (e.g. upgrade of farm, move of tenant, adding a tenant, load balancing of the data, and the like). When tenant data is needed for an operation (e.g. restore), the history may be accessed to determine the location of the data. More details regarding the backup manager are disclosed below.

[0019] FIGURE 2 shows a system for maintaining a location of tenant data across tenant moves. As illustrated, system 200 includes service 210, data store 220, data store 230 and computing device 240.

[0020] The computing devices used may be any type of computing device that is configured to perform the operations relating to the use of the computing device. For example, some of the computing devices may be: mobile computing devices (e.g. cellular phones, tablets, smart phones, laptops, and the like); some may be desktop computing devices and other computing devices may be configured as servers. Some computing devices may be arranged to provide an online cloud based service (e.g. service 210), some may be arranged as data shares that provide data storage services, some may be arranged in local networks, some may be arranged in networks accessible through the Internet, and the like.

[0021] The computing devices are coupled through network 18. Network 18 may be many different types of networks. For example, network 18 may be an IP network, a carrier network for cellular communications, and the like. Generally, network 18 is used to transmit data between computing devices, such as computing device 240, data store 220, data store 230 and service 210.

[0022] Computing device 240 includes application 242, Web browser 244 and user interface 246. As illustrated, computing device 240 is used by a user to interact with a service, such as service 210. According to an embodiment, service 210 is a multi-tenancy service. Generally, multi-tenancy refers to the isolation of data (including backups), usage

and administration between customers. In other words, data from one customer (tenant 1) is not accessible by another customer (tenant 2) even though the data from each of the tenants may be stored within a same database within the same data store.

5 **[0023]** User interface (UI) 246 is used to interact with various applications that may be local/non-local to computing device 240. One or more user interfaces of one or more types may be used to interact with the document. For example, UI 246 may include the use of a context menu, a menu within a menu bar, a menu item selected from a ribbon user interface, a graphical menu, and the like. Generally, UI 246 is configured such that a user may easily interact with functionality of an application. For example, a user may simply
10 select an option within UI 246 to select to restore tenant data that is maintained by service 210.

[0024] Data store 220 and data store 230 are configured to store tenant data. The data stores are accessible by various computing devices. For example, the network stores may be associated with an online service that supports online business point of solution
15 services. For example, an online service may provide data services, word processing services, spreadsheet services, and the like.

[0025] As illustrated, data store 220 includes tenant data, including corresponding backup data, for N different tenants. A data store may store all/portion of a tenant's data. For example, some tenants may use more than one data store, whereas other tenants share
20 the data store with many other tenants. While the corresponding backup data for a tenant is illustrated within the same data store, the backup data may be stored at other locations. For example, one data store may be used to store tenant data and one or more other data stores may be used to store the corresponding backup data.

[0026] Data store 230 illustrates a location of tenant data being changed and backup
25 data being changed from a different data store. In the current example, tenant data 2 and the corresponding backup data has been changed from data store 220 to data store 230. Backup data for tenant 3 has been changed from data store 220 to data store 230. Tenant data 8 has been changed from data store 220 to data store 230. The location change may occur for a variety of reasons. For example, more space may be needed for a tenant, the
30 data stores may be load balanced, the farm where the tenant is located may be upgraded, a data store may fail, a database may be moved/upgraded, and the like. Many other scenarios may cause a tenant's data to be changed. As can be seen from the current example, the tenant's data may be stored in one data store and the corresponding backup data may be stored in another data store.

[0027] Service 210 includes backup manager 26, history 212 and Web application 214 that comprises Web renderer 216. Service 210 is configured as an online service that is configured to provide services relating to displaying an interacting with data from multiple tenants. Service 210 provides a shared infrastructure for multiple tenants.

5 According to an embodiment, the service 210 is MICROSOFT'S SHAREPOINT ONLINE service. Different tenants may host their Web applications/site collections using service 210. A tenant may also use a dedicated alone or in combination with the services provided by service 210. Web application 214 is configured for receiving and responding to requests relating to data. For example, service 210 may access a tenant's data that is
10 stored on network store 220 and/or network store 230. Web application 214 is operative to provide an interface to a user of a computing device, such as computing device 240, to interact with data accessible via network 18. Web application 214 may communicate with other servers that are used for performing operations relating to the service.

[0028] Service 210 receives requests from computing devices, such as computing
15 device 240. A computing device may transmit a request to service 210 to interact with a document, and/or other data. In response to such a request, Web application 214 obtains the data from a location, such as network share 230. The data to display is converted into a markup language format, such as the ISO/IEC 29500 format. The data may be converted by service 210 or by one or more other computing devices. Once the Web application 214
20 has received the markup language representation of the data, the service utilizes the Web renderer 216 to convert the markup language formatted document into a representation of the data that may be rendered by a Web browser application, such as Web browser 244 on computing device 240. The rendered data appears substantially similar to the output of a corresponding desktop application when utilized to view the same data. Once Web
25 renderer 216 has completed rendering the file, it is returned by the service 210 to the requesting computing device where it may be rendered by the Web browser 244.

[0029] The Web renderer 216 is also configured to render into the markup language file one or more scripts for allowing the user of a computing device, such as computing device 240 to interact with the data within the context of the Web browser 244. Web
30 renderer 216 is operative to render script code that is executable by the Web browser application 244 into the returned Web page. The scripts may provide functionality, for instance, for allowing a user to change a section of the data and/or to modify values that are related to the data. In response to certain types of user input, the scripts may be executed. When a script is executed, a response may be transmitted to the service 210

indicating that the document has been acted upon, to identify the type of interaction that was made, and to further identify to the Web application 214 the function that should be performed upon the data.

[0030] In response to an operation that causes a change in location of a tenant's data, backup manager 26 places an entry into history 212. History 212 maintains a record of the locations for the tenant's data and corresponding backup data. According to an embodiment, history 212 stores the database name and location that is used to store the tenant's data, a name and location of the backup location for the tenant's data and the time the data is stored at that location (See FIGURE 3 and related discussion). The history information may be stored in a variety of ways. For example, history records for each tenant may be stored within a database, history information may be stored within a data file, and the like.

[0031] According to an embodiment, backup manager 26 is configured to perform full backups of tenant data and incremental backups and transaction log entries between the times of the full backups. The scheduling of the full backups is configurable. According to an embodiment, full backups are performed weekly, incremental backups are performed daily and transactions are stored every five minutes. Other schedules may also be used and may be configurable. The different backups may be stored in a same locations and/or different locations. For example, full backups may be stored in a first location and the incremental and transaction logs may be stored in a different location.

[0032] FIGURE 3 shows a history including records for tenant data location changes. History 300 includes records for each tenant that is being managed. For example purposes, history 300 shows history records for Tenant 1 (310), Tenant 2 (320) and Tenant 8 (330).

[0033] As illustrated, history record 310 was created in response to Tenant 1 being added. According to an embodiment, a history record comprises fields for a content location, a time, a backup location and a time. The content location provides information on where the tenant's content is stored (e.g. a database name, a URL to the content location, and the like). The Time1 field indicates a last time the tenant's data was at the specified location. According to an embodiment, when the Time1 field is empty, the Time2 value is used for the record. When the Time1 field and the Time2 field are both empty, the data is still located at the content location and the backup location listed in the record. The backup location field specifies a location of where the backup for the content

is located. The Time2 field specifies a last time the tenant's backup data was at the specified location.

[0034] Referring to the history for Tenant1 (310) it can be seen that Tenant 1's data is located at content location "Content12" (e.g. a name of a database) and that the backup data for Tenant 1's data is located at "backups\ds220\Content 12." In this case, Tenant 1's data has not changed location since Tenant 1 was added.

[0035] Tenant 2's data has changed locations from "Content 12" to "Content 56" to "Content 79." Before 3-4-2010 at 10AM and after 1-2-2010 at 1:04AM the data is stored at "Content 56" and the corresponding backup data is stored at "backups\ds220\Content 56." Before 1-2-2010 at 1:04AM the data is stored at "Content 12" and the corresponding backup data is stored at "backups\ds220\Content 12."

[0036] Tenant 3's data has changed locations from "Content 12" to "Content 15." The corresponding backup data has changed from "backups\ds220\Content 12" to "backups\ds220\Content 15" to "backups\ds230\Content 79." Tenant's 3 data is stored at "Content 15" after 3-12-2010 at 7:35AM. Before 3-24-2010 at 1:22AM and after 3-12-2010 at 7:35AM the corresponding backup data is stored at "backups\ds220\Content 15." Before 3-12-2010 at 7:35AM the data is stored at "Content 12" and the corresponding backup data is stored at "backups\ds220\Content 12." In the current example, Tenant 3's location of the backup data changed without changing the location of the Tenant data from "Content 15."

[0037] Many other ways may be used to store the information relating to the location of tenant data. For example, the time field could include a start time and an end time, a start time and no end time, or an end time and no start time. The location could be specified as a name, an identifier, a URL, and the like. Other fields may also be included, such as a size field, a number of records field, a last accessed field, and the like.

[0038] FIGURES 4 and 5 show an illustrative process for recovering tenant data across tenant moves. When reading the discussion of the routines presented herein, it should be appreciated that the logical operations of various embodiments are implemented (1) as a sequence of computer implemented acts or program modules running on a computing system and/or (2) as interconnected machine logic circuits or circuit modules within the computing system. The implementation is a matter of choice dependent on the performance requirements of the computing system implementing the invention. Accordingly, the logical operations illustrated and making up the embodiments described herein are referred to variously as operations, structural devices, acts or modules. These

operations, structural devices, acts and modules may be implemented in software, in firmware, in special purpose digital logic, and any combination thereof.

[0039] FIGURE 4 illustrates a process for updating a history of a tenant's data location change.

5 **[0040]** After a start block, process 400 moves to operation 410, where a determination is made that an operation has changed a location of a tenant's data. The change may relate to all/portion of a tenant's data. Many different operations may cause a change in a location of tenant data. For example, adding a tenant, farm upgrade, moving of tenant, load balancing the tenant's data, load balancing the corresponding backup data,
10 a maintenance operation, a failure, and the like. Generally, any operation that causes the tenant's data and/or corresponding backup data to change locations is determined.

[0041] Flowing to operation 420, the history for the tenant whose data is changing location is accessed. The history may be accessed within a local data store, a shared data store and/or some other memory location.

15 **[0042]** Moving to operation 430, the history for the tenant is updated to reflect a current state and any previous states of the tenant's data. According to an embodiment, each tenant includes a table indicating its corresponding history. The history may be stored using many different methods using many different types of structures. For example, the history may be stored in a memory, a file, a spreadsheet, a database, and the
20 like. History records may also be intermixed within a data store, such as within a list, a spreadsheet and the like. According to an embodiment, a history record comprises fields for a content location, a time, a backup location and a time. The content location provides information on where the tenant's content is stored (e.g. a database name, a URL to the content location, and the like). The Time1 field indicates a last time the tenant's data was
25 at the specified location. According to an embodiment, when the Time1 field is empty, the Time1 value is the same as the Time2 field. When the Time1 field and the Time2 field are empty, the data is still located at the content location and the backup location. The backup location field specifies a location of where the backup for the content is located. The Time2 field specifies a last time the tenant's backup data was at the specified location.

30 **[0043]** The process then flows to an end block and returns to processing other actions.

[0044] FIGURE 5 shows a process for processing a request for restoring tenant data from a previous location.

2012238127 28 Oct 2016

- 10 -

[0045] After a start block, the process moves to operation 510, where a request is received to restore tenant data. For example, a tenant may have accidentally deleted data that they would like to restore. According to an embodiment, the request includes a time indicating when they believe that they deleted the data. According to another embodiment, a time range may be given. According to yet another embodiment, each location within the tenant's history may be searched for the data without providing a time within the request.

[0046] Flowing to operation 520, the history for the tenant is accessed to determine where the data is located. As discussed above, the history includes a current location of tenant data and corresponding backup data and each of the previous locations of the data.

[0047] Moving to operation 530, the tenant's data is restored to a temporary location such that the tenant's current data is not overwritten with unwanted previous data.

[0048] Transitioning to operation 540, the requested data is extracted from the temporary location and restored to the current location of the tenant's data. The data at the temporary location may be erased.

[0049] The process then flows to an end block and returns to processing other actions.

[0050] The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

[0051] Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

[0052] The reference in this specification to any prior publication (or information derived from it), or to any matter which is known, is not, and should not be taken as an acknowledgment or admission or any form of suggestion that that prior publication (or information derived from it) or known matter forms part of the common general knowledge in the field of endeavour to which this specification relates.

2012238127 28 Oct 2016

- 11 -

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1: A method executing on a processor for recovering tenant data across tenant moves, comprising:

performing full backup operations on the tenant data to generate tenant backup data according to a full backup schedule;

performing incremental backup operations on the tenant data to generate the tenant backup data according to an incremental backup schedule;

determining an operation that changes a storage location of where a tenant's data is stored within a network storage location to a different network storage location;

in response to the operation that changes the storage location of the tenant's data, updating a history for the tenant of the storage location of the tenant's data by adding a current storage location of where the tenant's data is currently stored, wherein the history identifies the current storage location and a time associated with when the tenant's data was stored at the current storage location and a backup location and a time associated with when the tenant's backup data was stored at the backup location;

when requested, accessing the history for the tenant to determine a previous storage location of where the tenant's data was stored based at least in part on a time indicating when the tenant's data was deleted; and

restoring the tenant's data using the previous storage location.

2: The method of Claim 1, wherein the history is updated in response to a load balancing of at least one of: tenant data and backup data.

3: The method of Claim 1, wherein the history is updated in response to a tenant move.

4: The method of Claim 1, wherein the history is updated in response to a farm upgrade.

2012238127 28 Oct 2016

- 12 -

5: The method of Claim 1, wherein updating the history comprises storing a storage location of backup data that corresponds to a backup of the tenant's data.

6: The method of Claim 5, wherein the backup data comprises a full backup of the tenant's data and incremental backups of the tenant's data and transaction log backups of the tenant's data.

7: The method of Claim 1, wherein updating the history comprises including a time indicating when the tenant's data is moved from the previous storage location the current storage location.

8: The method of Claim 7, further comprising determining the previous storage location of the tenant's data by accessing the storage location based upon a comparison of a specified time with the time within the history.

9: The method of Claim 1, further comprising restoring the data to a temporary storage location and extracting requested data from the temporary storage location and placing the extracted data into the current storage location of the tenant's data.

10: A computer-readable storage medium, excluding a signal, storing computer-executable instructions for recovering tenant data across tenant moves, comprising:

performing full backup operations on the tenant data to generate tenant backup data according to a full backup schedule;

performing incremental backup operations on the tenant data to generate the tenant backup data according to an incremental backup schedule;

determining an operation that changes a storage location of a tenant's data from a network storage location to a different network storage location;

updating a history for the tenant to include a current storage location of the tenant's data, wherein the history for the tenant includes a record for each storage location at which the tenant's data has been stored and the current storage location, wherein each record comprises a tenant data storage location, a backup storage location for the tenant

2012238127 28 Oct 2016

- 13 -

backup data, and time information indicating when the data was at each of the storage locations;

when requested, accessing the history to determine a previous storage location of the tenant's data using a time indicating when the tenant's data was deleted; and

restoring the tenant's data using the previous storage location.

11: The computer-readable storage medium of Claim 10, wherein the history is updated in response to at least one of: a load balancing on at least one of: tenant data and backup data; a tenant move; and a farm upgrade.

12: The computer-readable storage medium of Claim 10, further comprising providing each storage location of the backup of tenant's data in response to the request.

13: The computer-readable storage medium of Claim 12, wherein the backup data comprises a full backup of the tenant's data, incremental backups of the tenant's data, and transaction log data.

14: The computer-readable storage medium of Claim 10, wherein updating the history comprises including a time indicating when the tenant's data is moved from the previous storage location to the current storage location.

15: The computer-readable storage medium of Claim 14, further comprising determining the previous storage location of the tenant's data by accessing the storage location based upon a comparison of a specified time with the time within the history.

16: The computer-readable storage medium of Claim 10, further comprising restoring the data to a temporary storage location and extracting requested data from the temporary storage location and placing the extracted data into the current storage location of the tenant's data.

2012238127 28 Oct 2016

- 14 -

17: A system for recovering tenant data across tenant moves, comprising:
a network connection that is configured to connect to a network;
a processor, memory, and a computer-readable storage medium;
an operating environment stored on the computer-readable storage medium and executing on the processor;
a data store storing tenant data that is associated with different tenants; and
a backup manager operating that is configured to perform actions comprising:

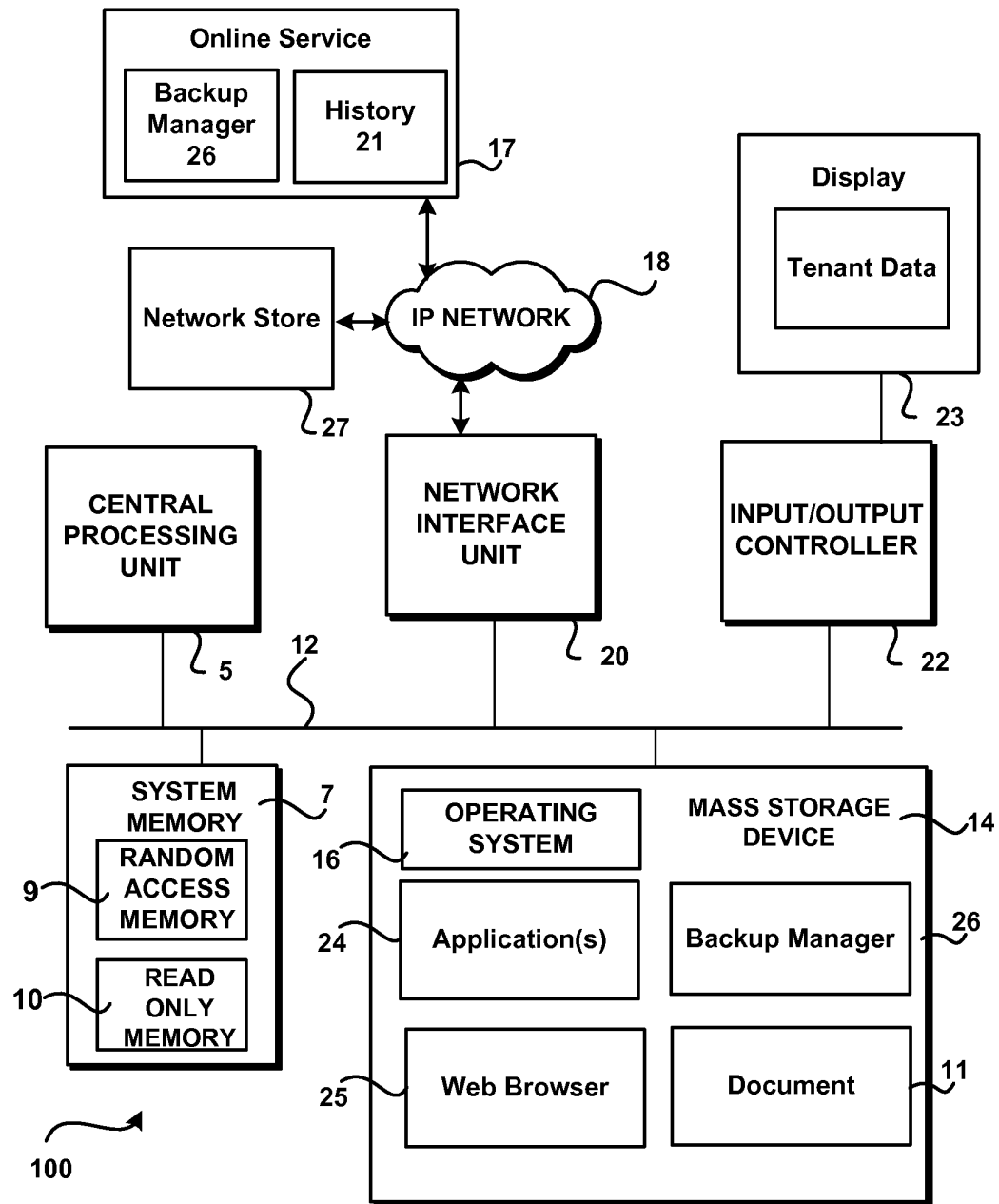
performing full backup operations on the tenant data to generate tenant backup data according to a full backup schedule;
performing incremental backup operations on the tenant data to generate the tenant backup data according to an incremental backup schedule
receiving a request for a tenant's data within a network data store;
accessing a history of the tenant including tenant data storage locations to determine a storage location of the requested tenant data using a time indicating when the tenant's data was moved to backup storage, wherein the history includes a record for each storage location at which the tenant's data has been stored and the current storage location, wherein the record comprises a tenant data storage location, a backup storage location for the tenant backup data, and time information indicating when the data was at each of the storage locations; and
restoring the tenant's data using the storage location.

18: The system of Claim 17, further comprising comparing a time specified within the request to determine the storage location of the requested tenant data.

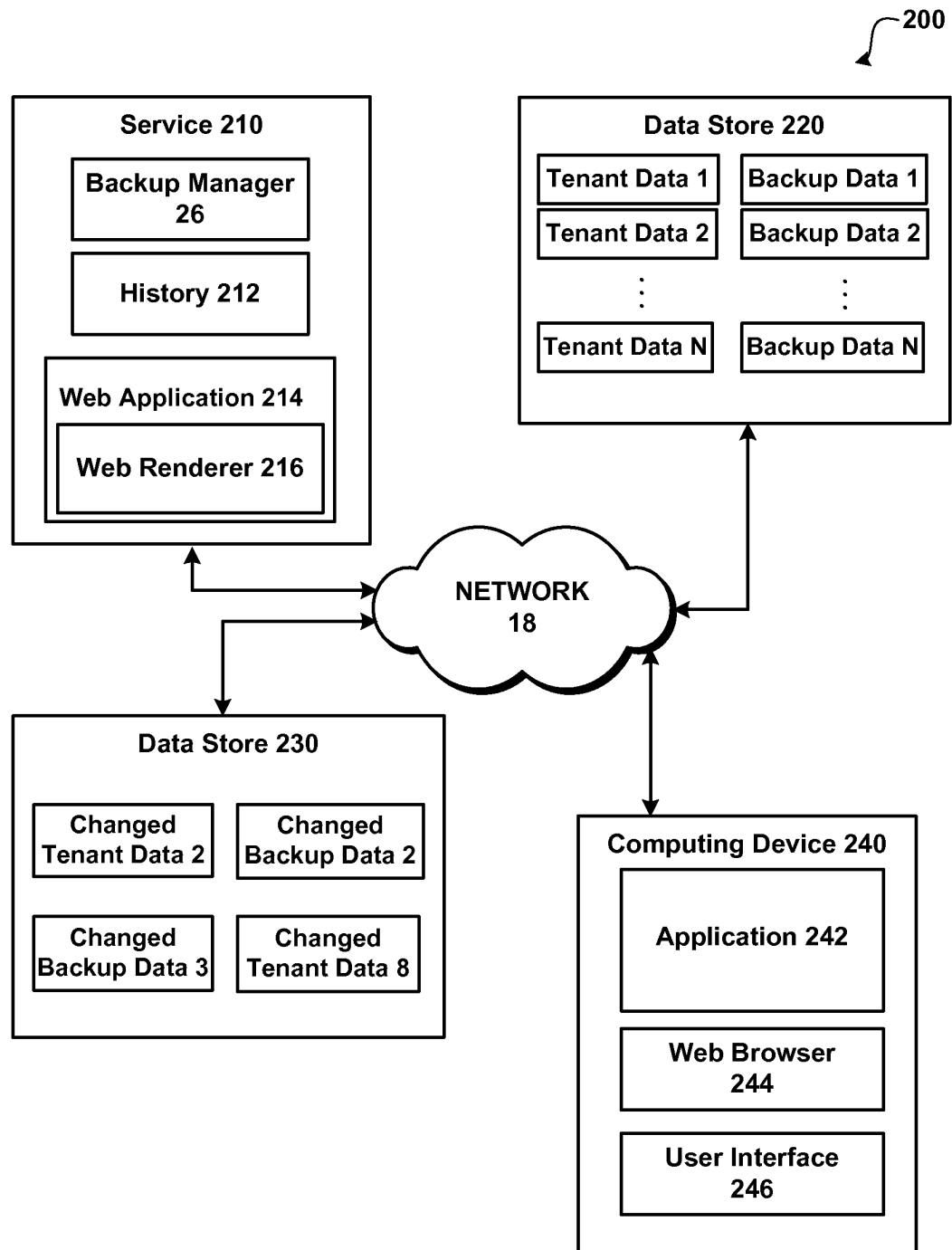
19: The system of Claim 17, further comprising examining each storage location within the history to determine the storage location of the requested tenant data.

20: The system of Claim 17, further comprising restoring the tenant's data to a temporary storage location and extracting the requested data from the temporary storage location and placing the extracted data into the current storage location of the tenant's data.

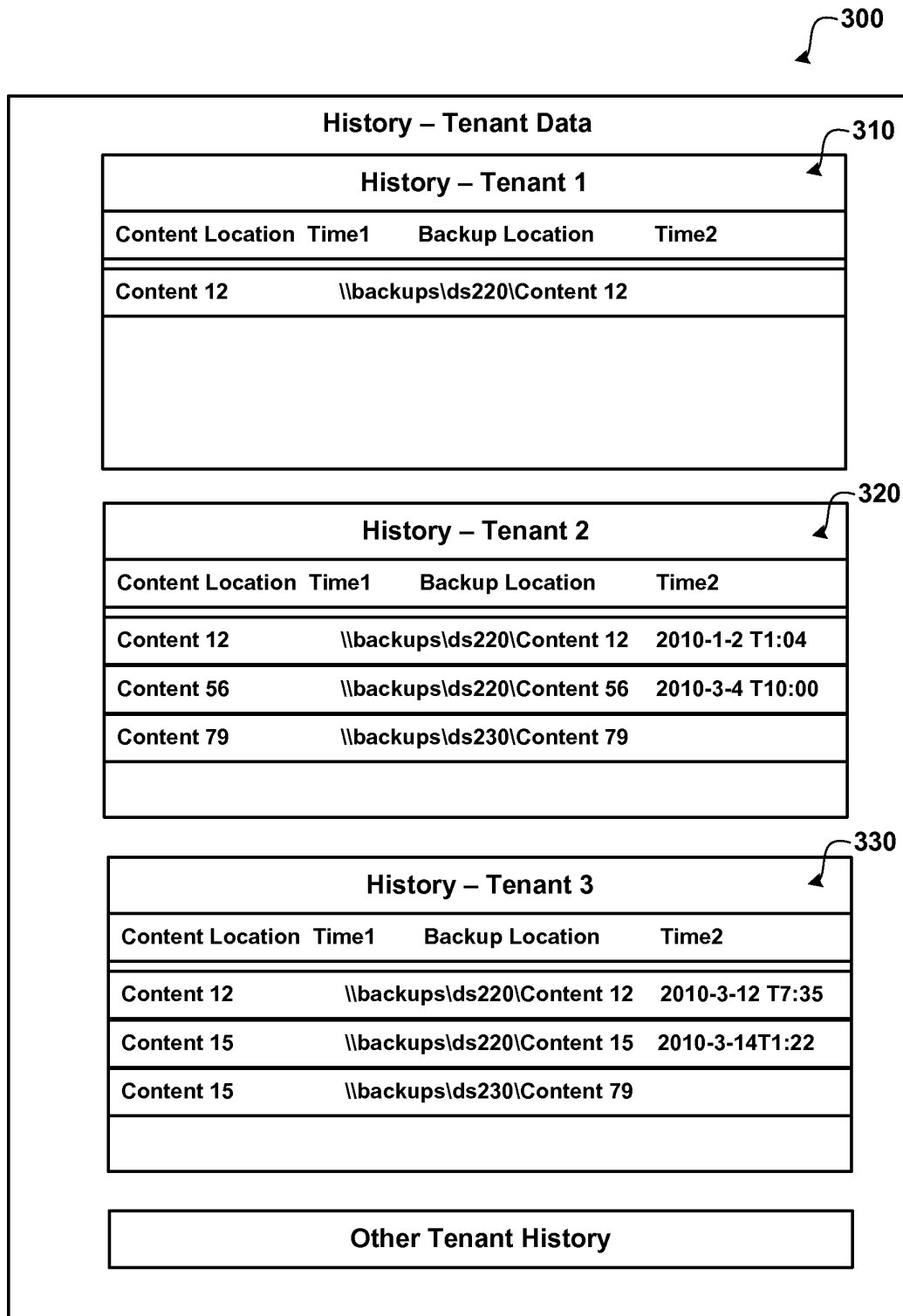
1/5

**FIG.1**

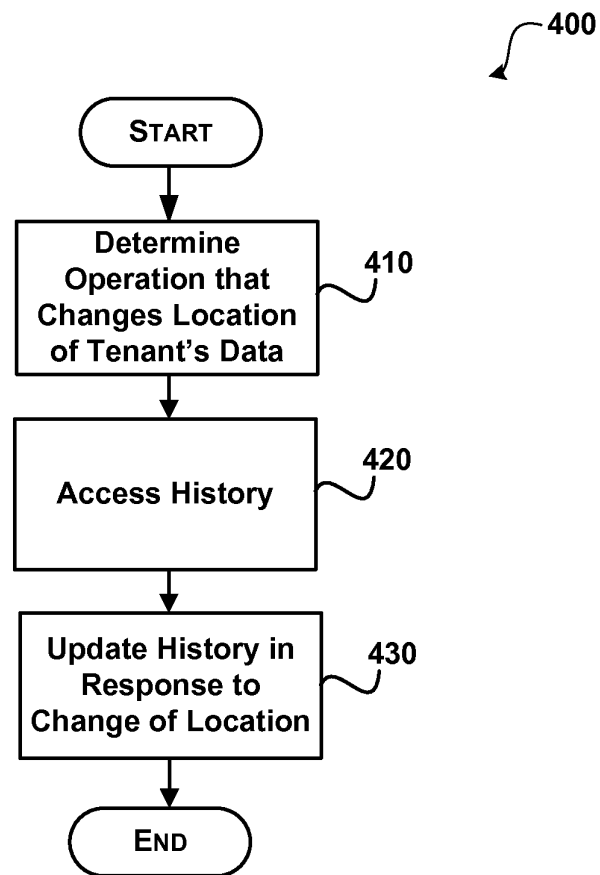
2/5

**FIG.2**

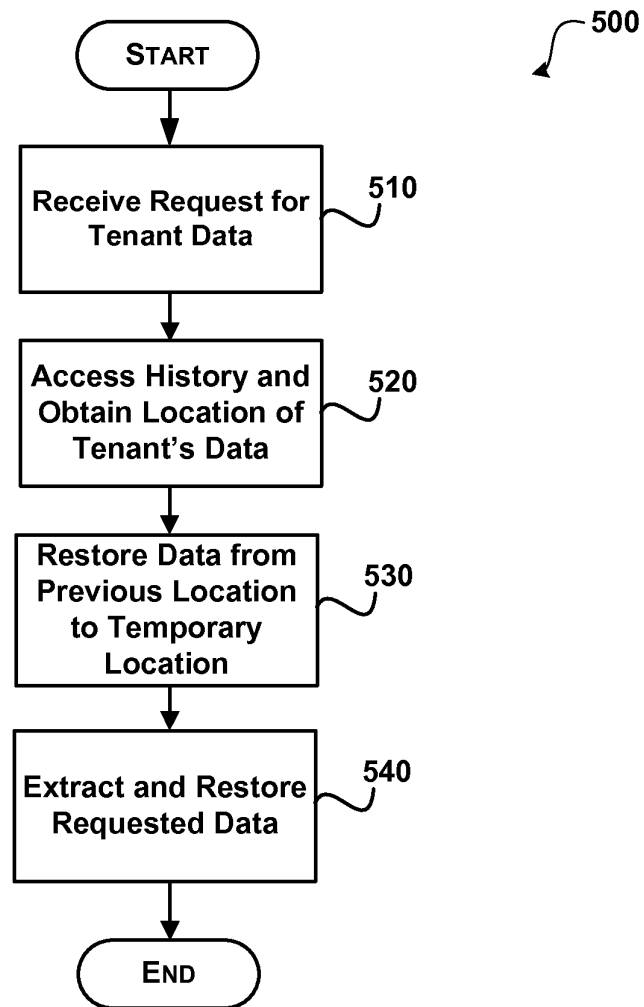
3/5

**FIG.3**

4/5

**FIG. 4**

5/5

**FIG.5**