



US 20190319841A1

(19) **United States**

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

(10) **Pub. No.: US 2019/0319841 A1**

(43) **Pub. Date: Oct. 17, 2019**

(54) **MULTI-REGION VIRTUAL DATA CENTER
TEMPLATE**

(60) Provisional application No. 62/078,367, filed on Nov.
11, 2014.

(71) Applicant: **Skytap**, Seattle, WA (US)

Publication Classification

(72) Inventors: **Bradley M. Schick**, Seattle, WA (US);
Sumit Mehrotra, Redmond, WA (US);
Ross Andrus, Sammamish, WA (US);
Matthew Hanlon, Seattle, WA (US);
Yelena Branch, Seattle, WA (US)

(51) **Int. Cl.**
H04L 12/24 (2006.01)

(52) **U.S. Cl.**
CPC **H04L 41/0843** (2013.01); **H04L 41/0813**
(2013.01)

(21) Appl. No.: **16/447,053**

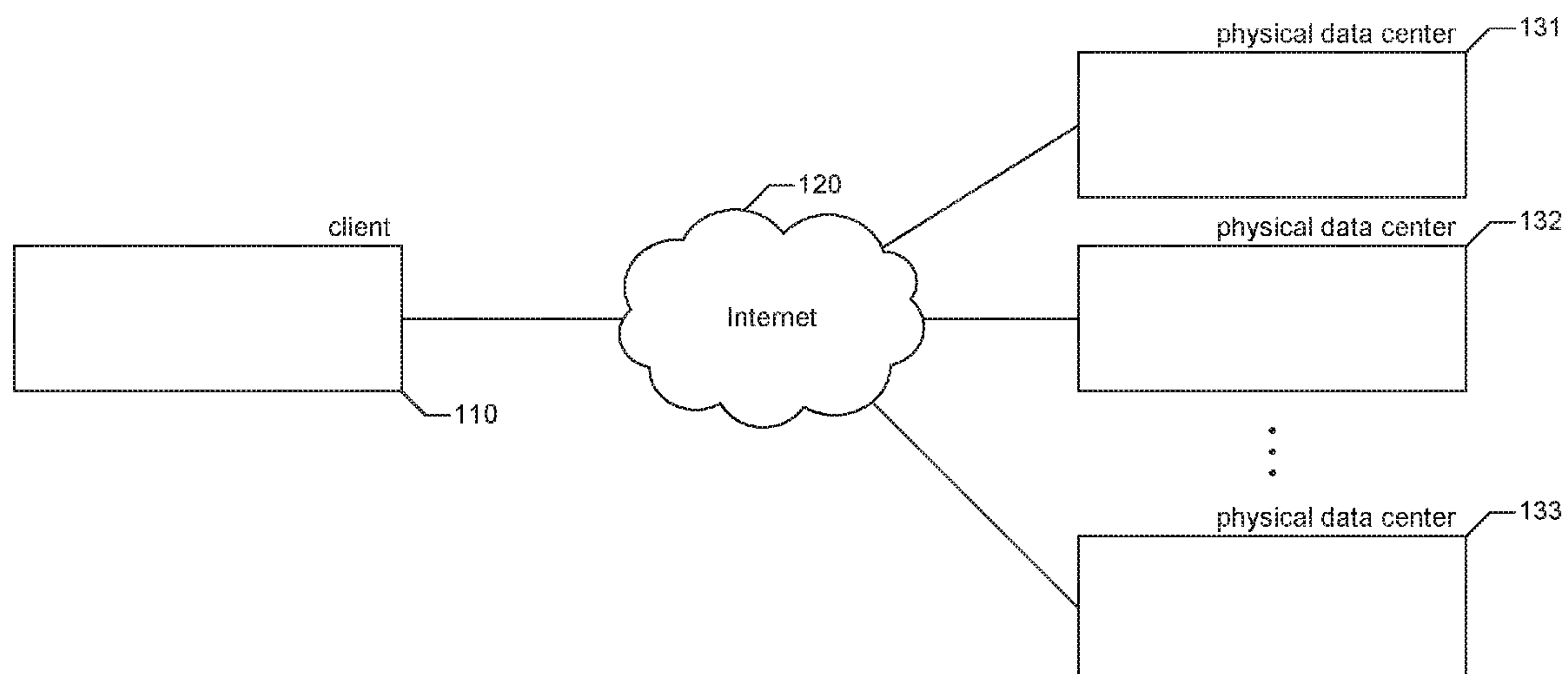
(57) **ABSTRACT**

(22) Filed: **Jun. 20, 2019**

Related U.S. Application Data

(63) Continuation of application No. 14/938,727, filed on
Nov. 11, 2015, now Pat. No. 10,374,891.

A facility for distributing a template recording a past state of
a virtual computing environment is described. The facility
accesses a list of physical data centers at which the template
is to be available for immediate use to instantiate a new
virtual computing environments. The facility causes the
template to be stored at each physical data center on the list.



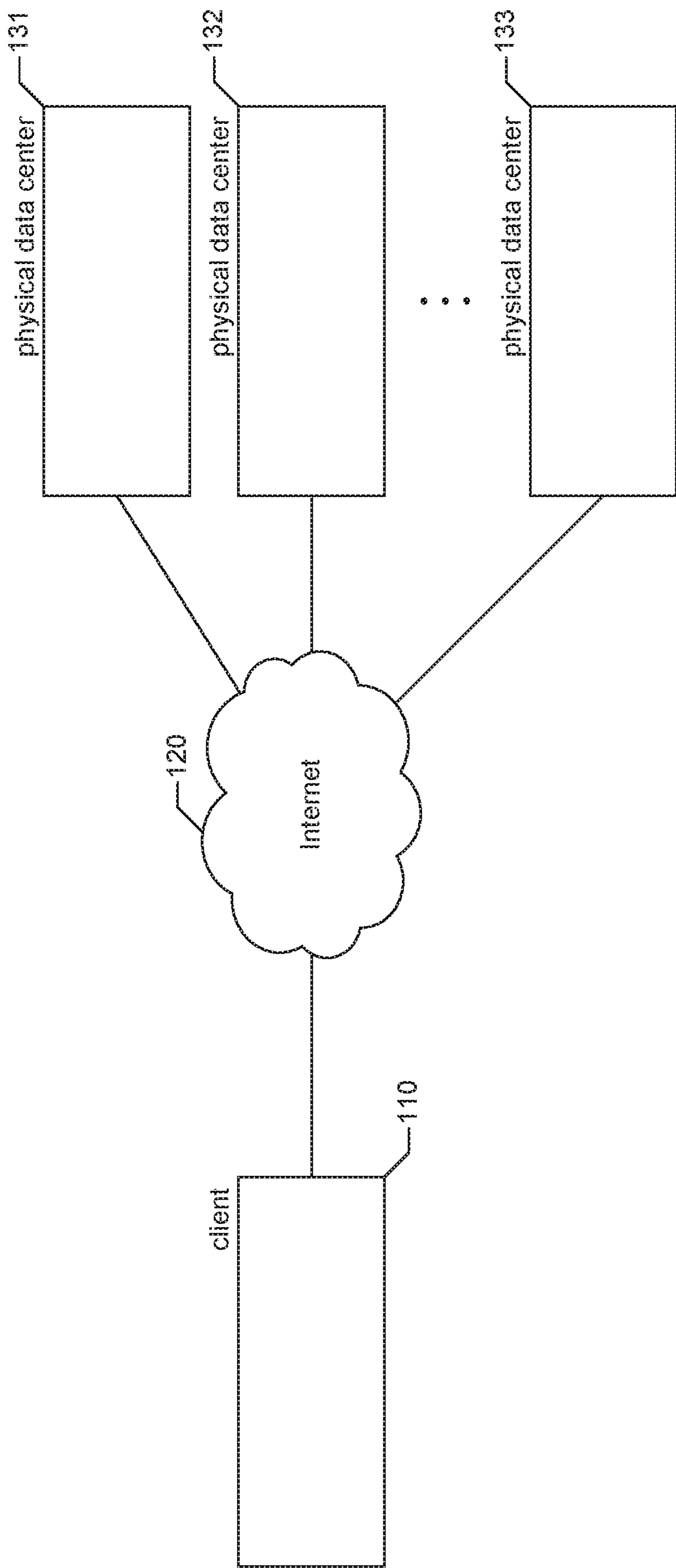
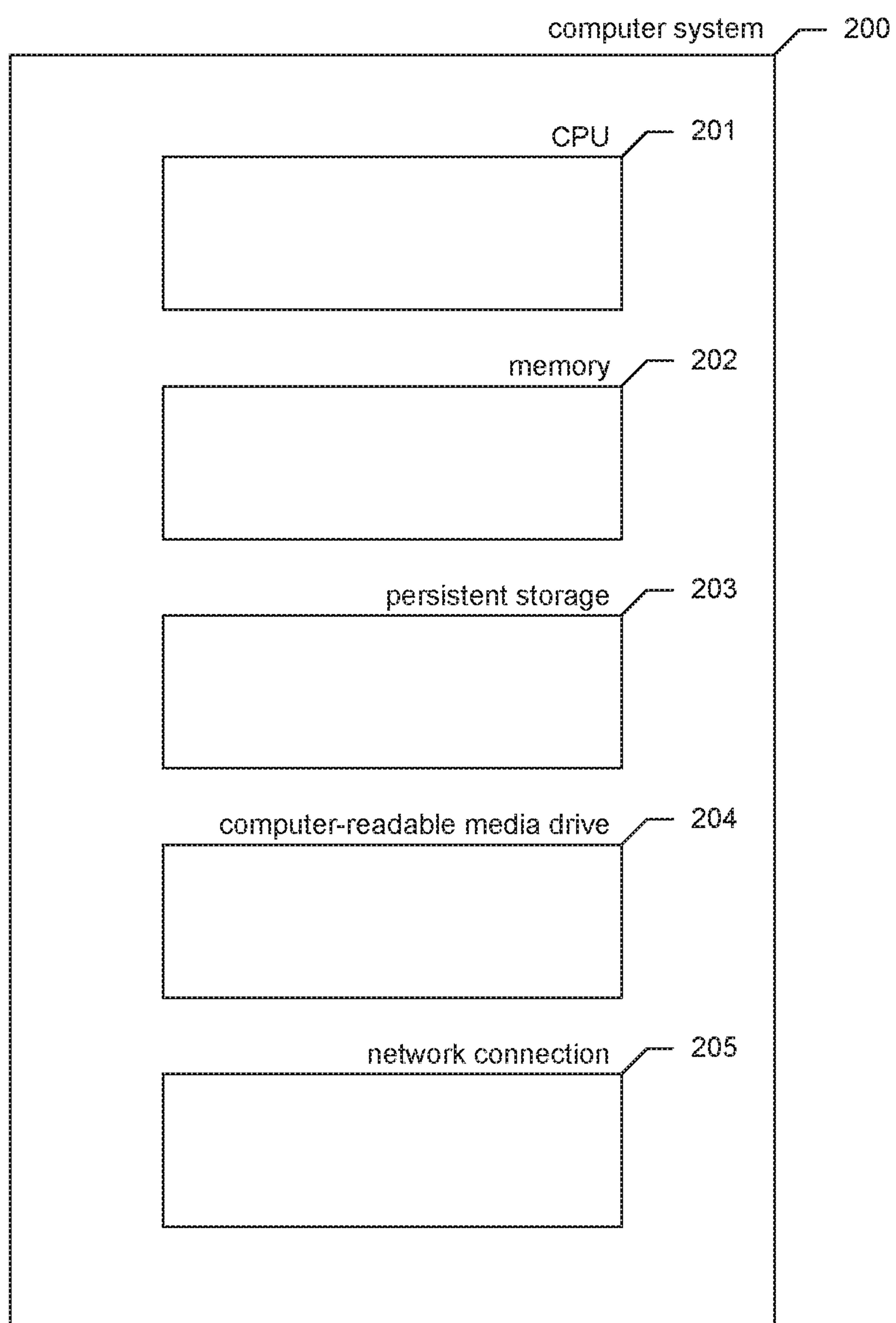


FIG. 1

**FIG. 2**

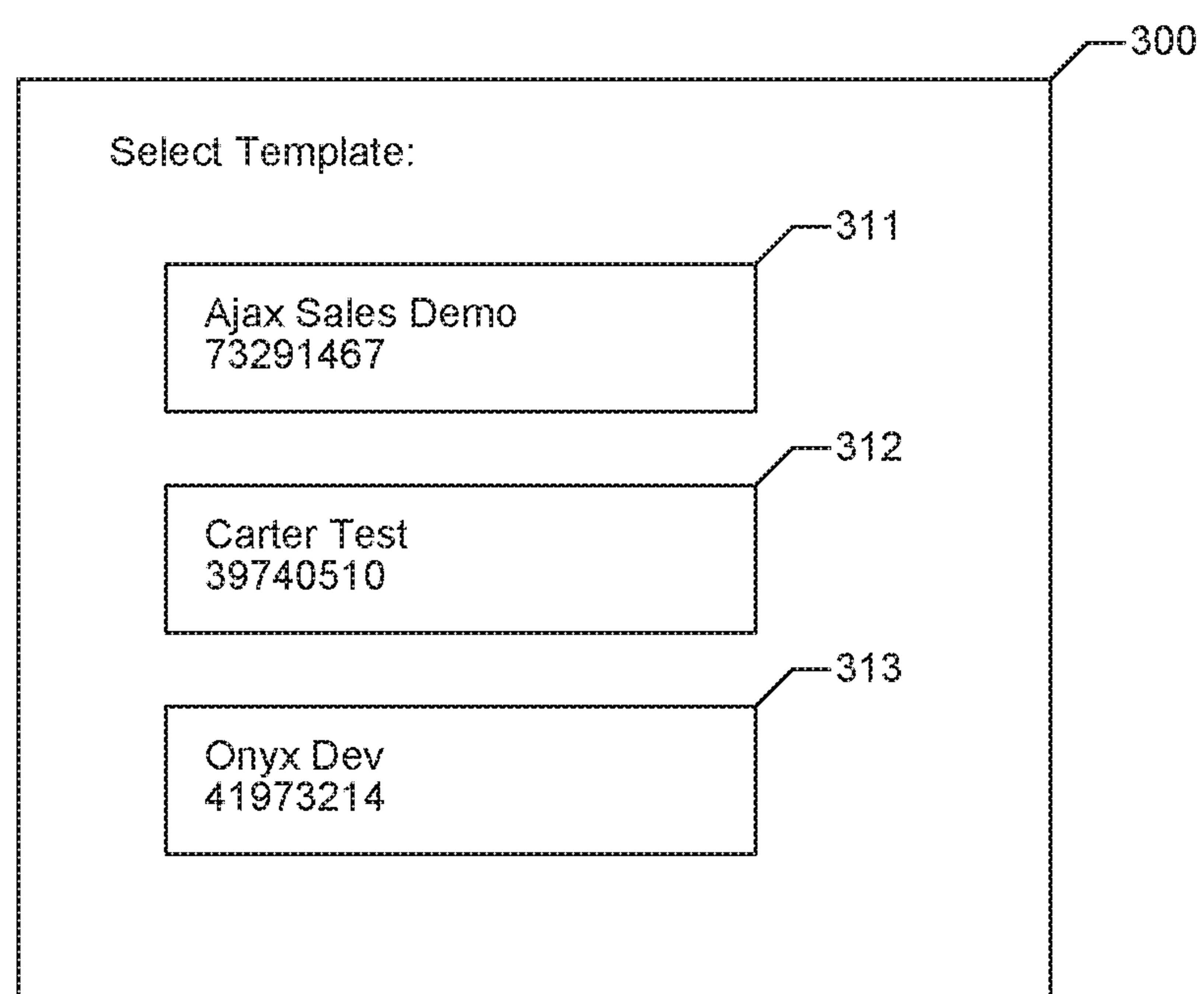


FIG. 3

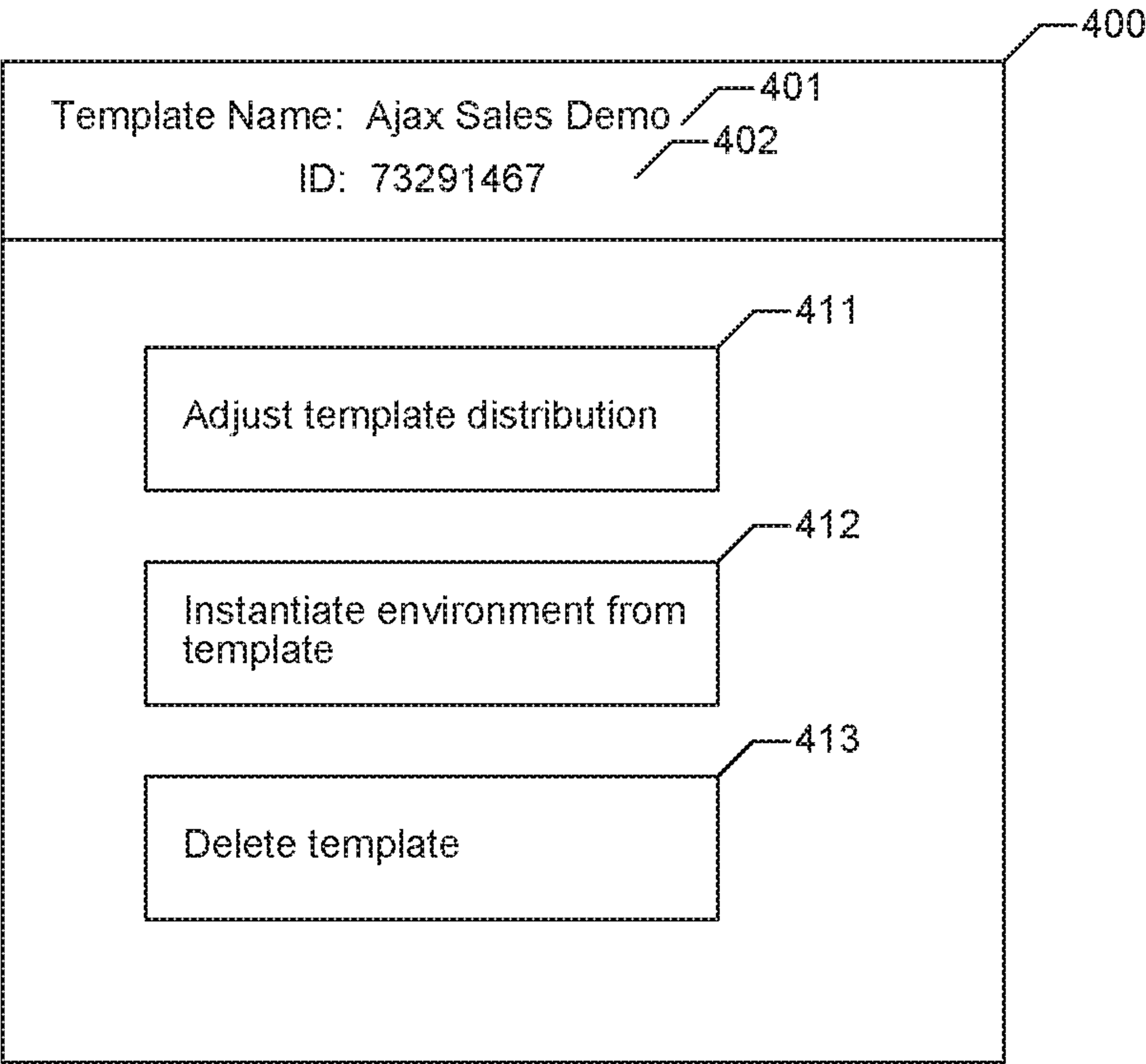
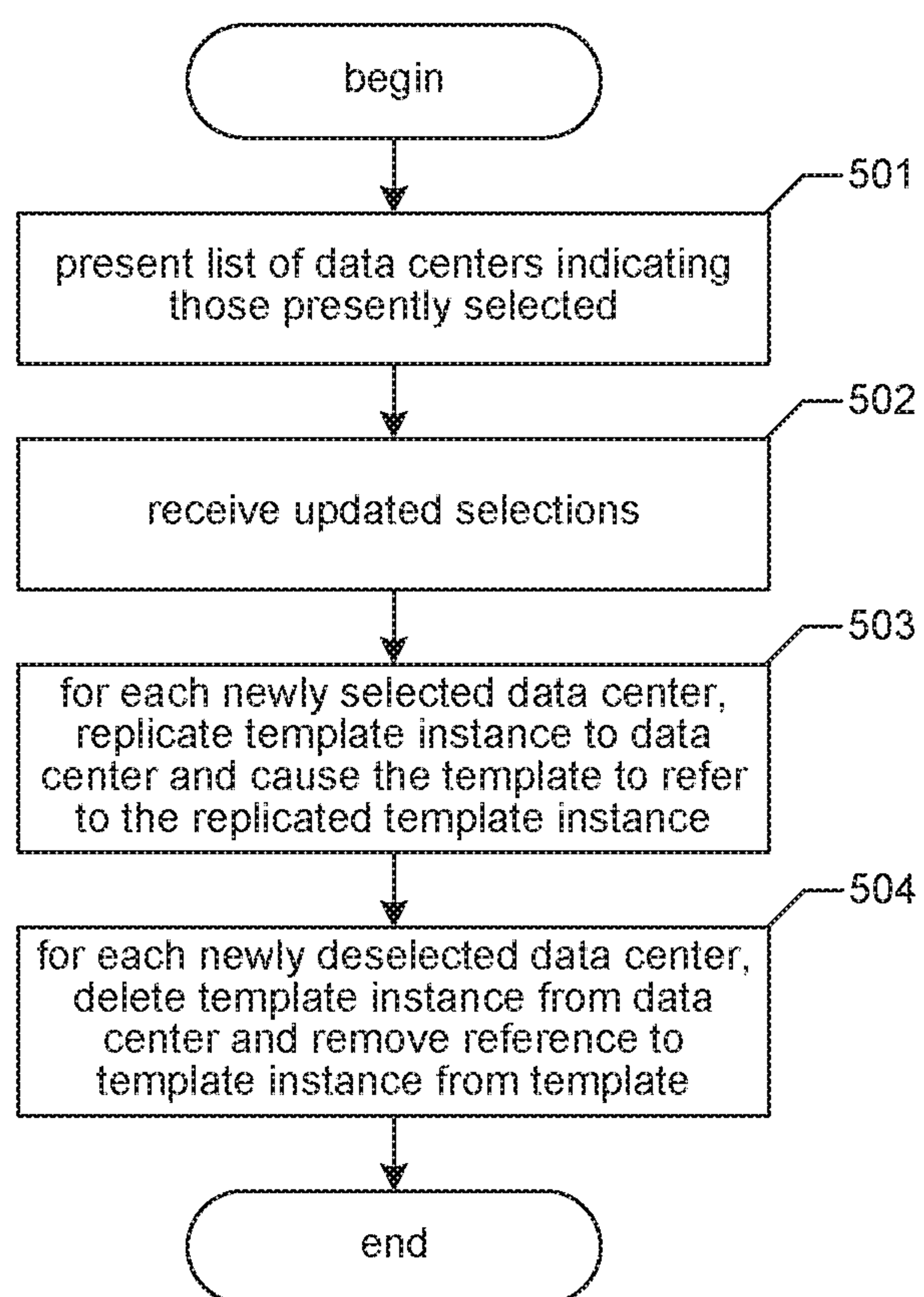


FIG. 4

**FIG. 5**

600

Template Name: Ajax Sales Demo601
ID: 73291467602

Replicate template among selected data centers:

☒ Beijing611
☒ Sydney612
☐ Alexandria613
☒ Chicago614
☐ San Francisco615
☐ Tokyo616
☐ Berlin617
☐ Dubai618

select all621

deselect all622

FIG. 6

templates table	
template id	template name
73291467	Ajax Sales Demo

FIG. 7A

template instances table				
template instance id	template id	location	creation time	pointer to saved template instance state
92837462	73291467	Beijing	1/15/2014 14:07:03.211	<pointer 1>
44318873	73291467	Sydney	2/01/2014 9:11:14.036	<pointer 2>
11062344	73291467	Chicago	2/01/2014 9:11:46.013	<pointer 3>

FIG. 7B

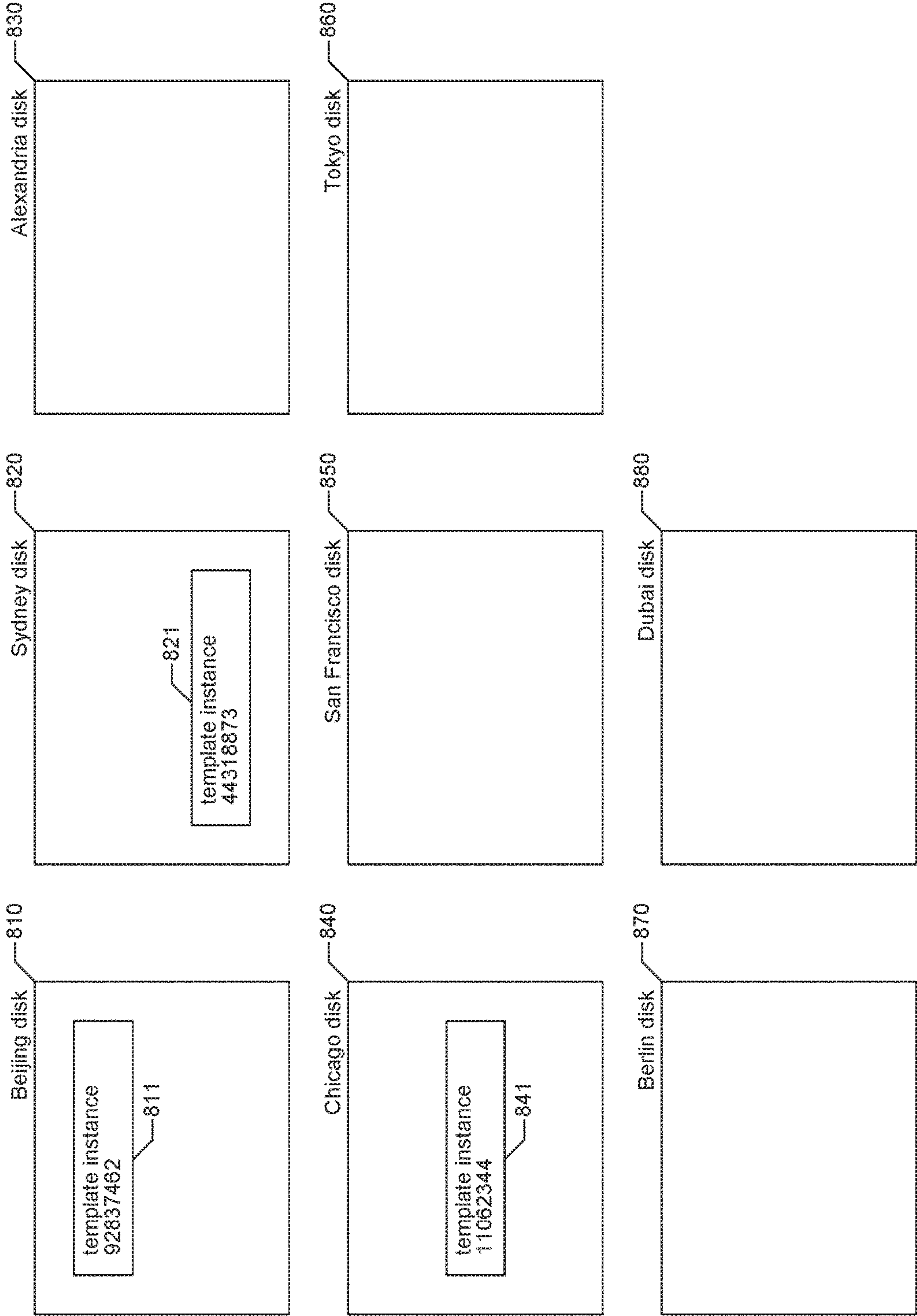
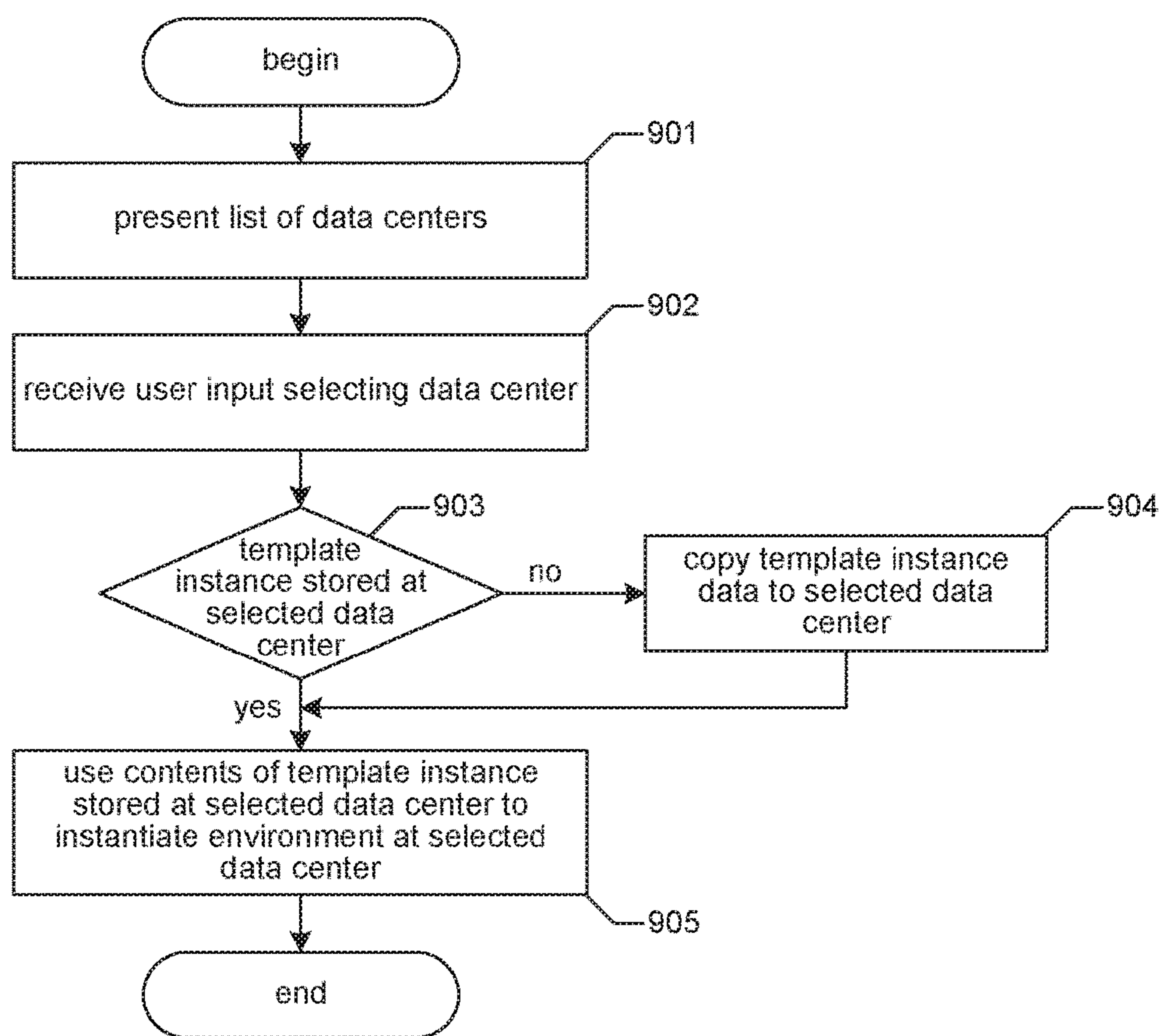


FIG. 8

**FIG. 9**

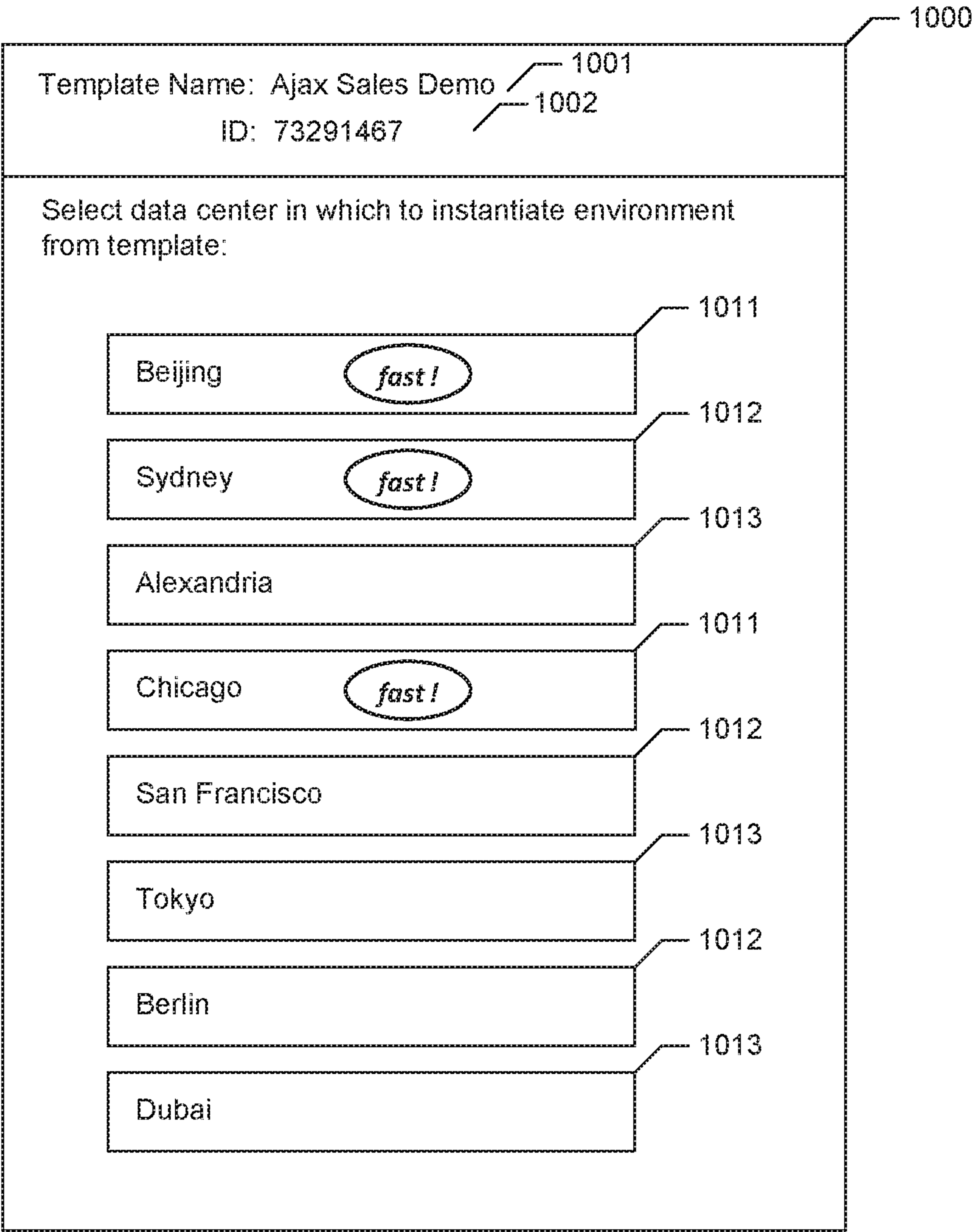


FIG. 10

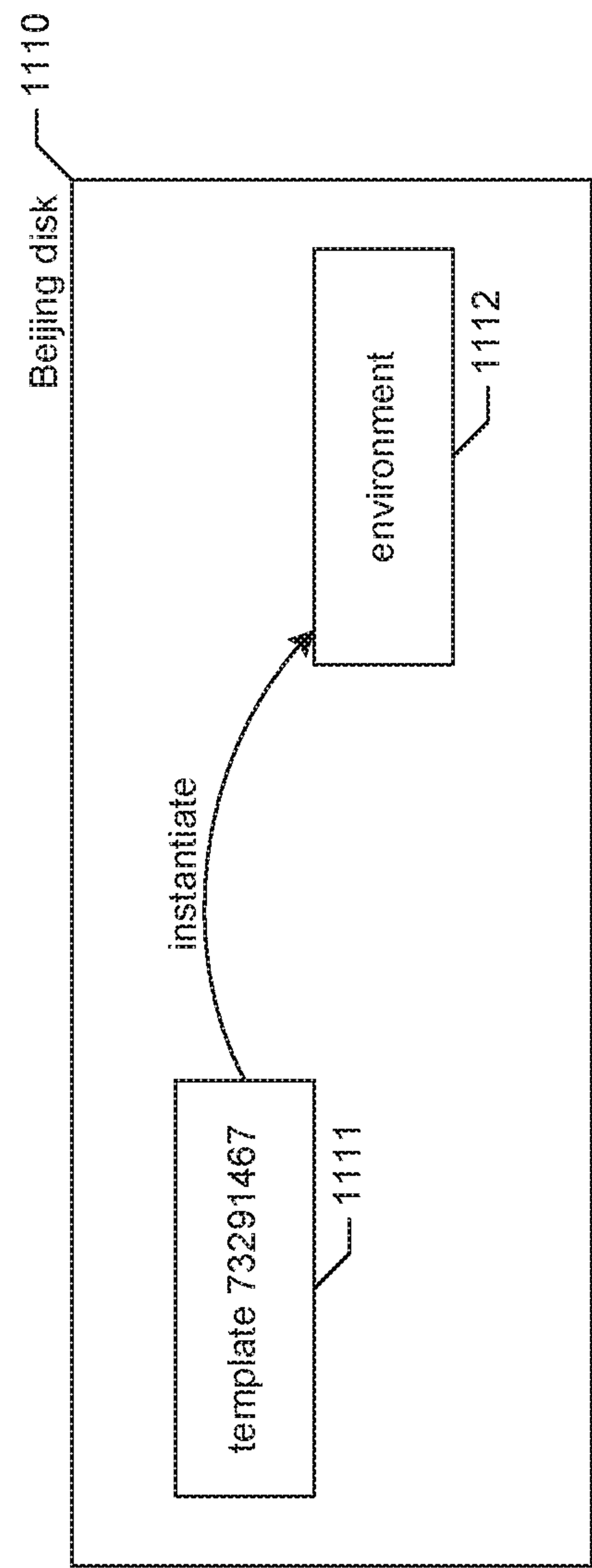


FIG. 11

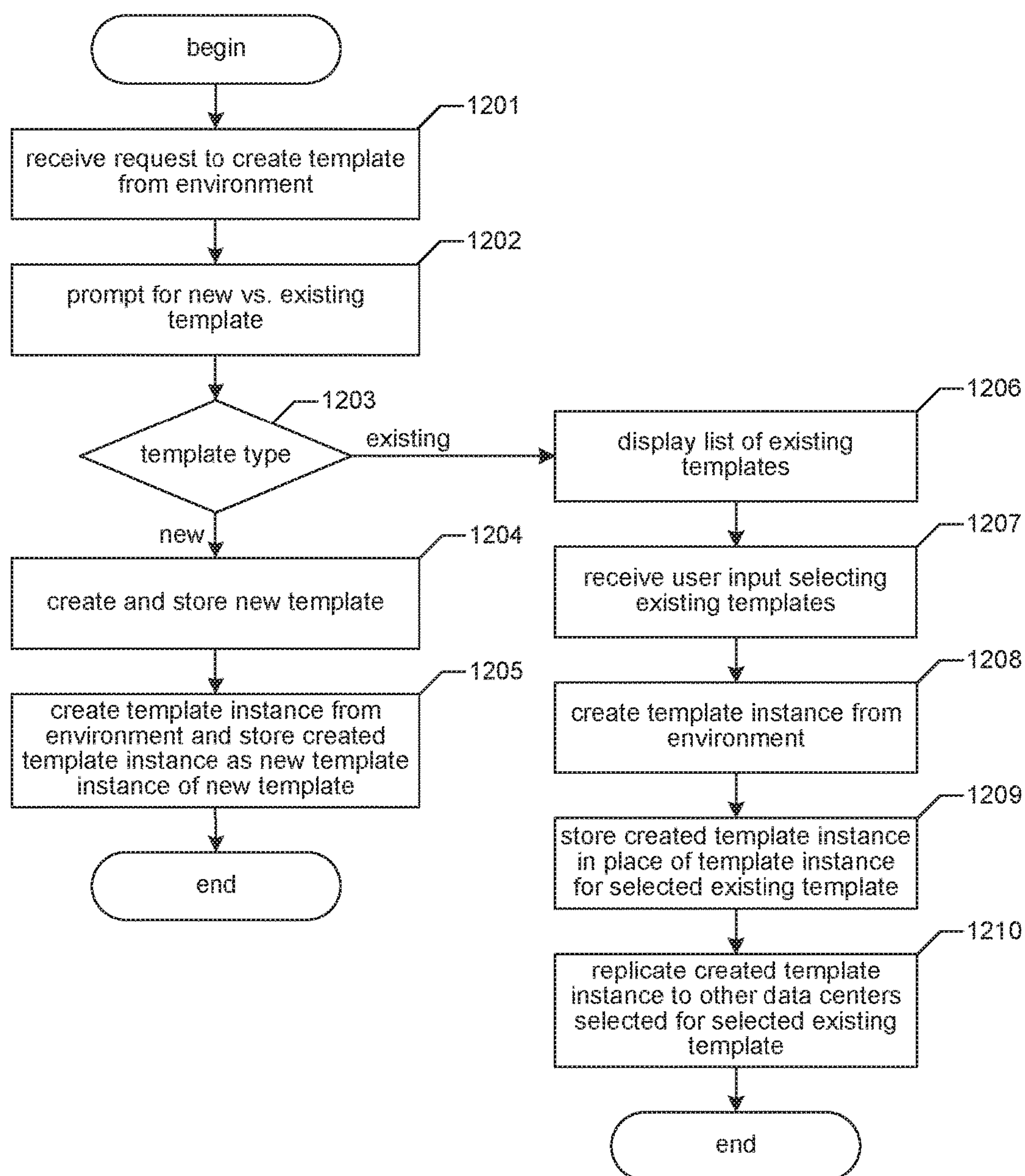


FIG. 12

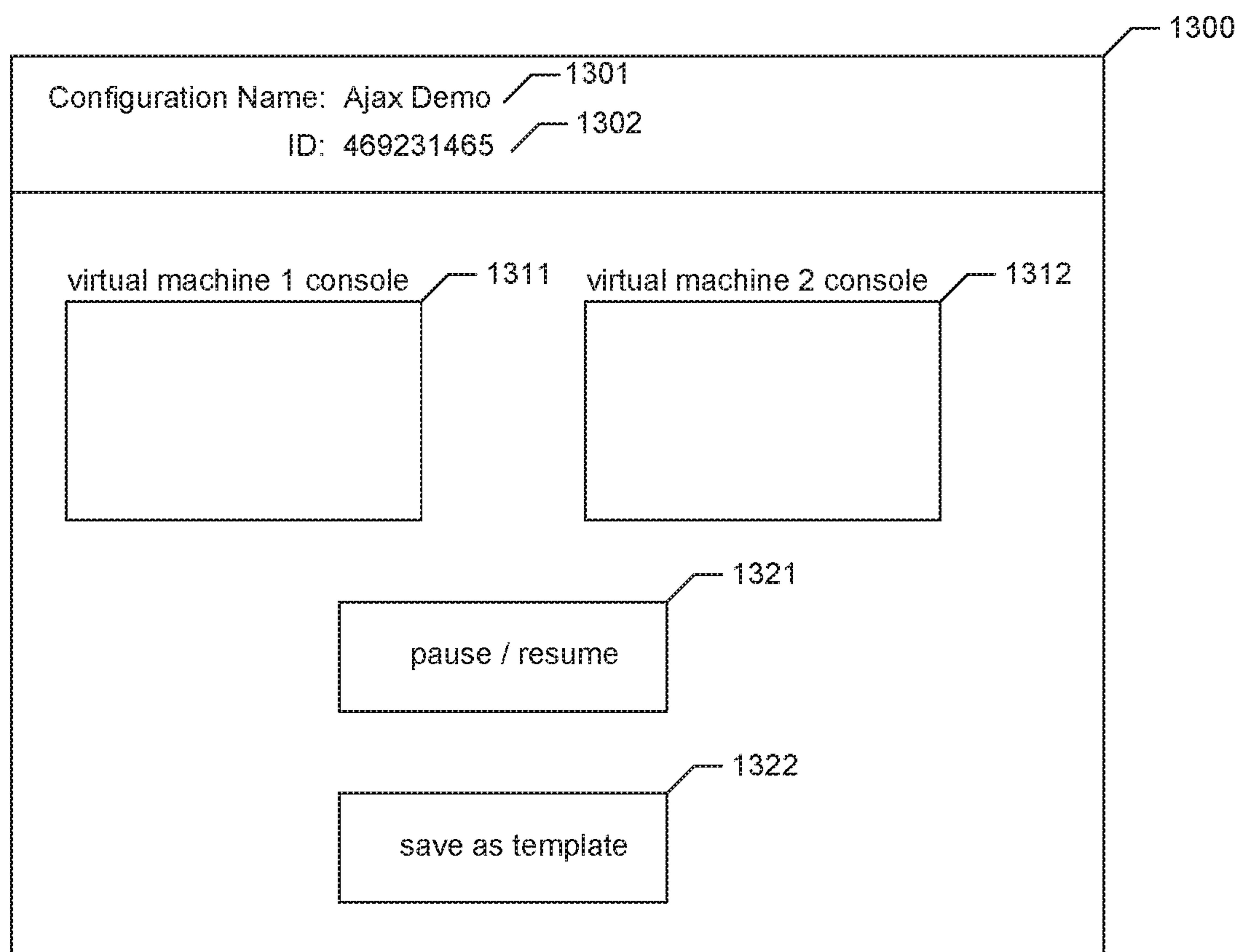


FIG. 13

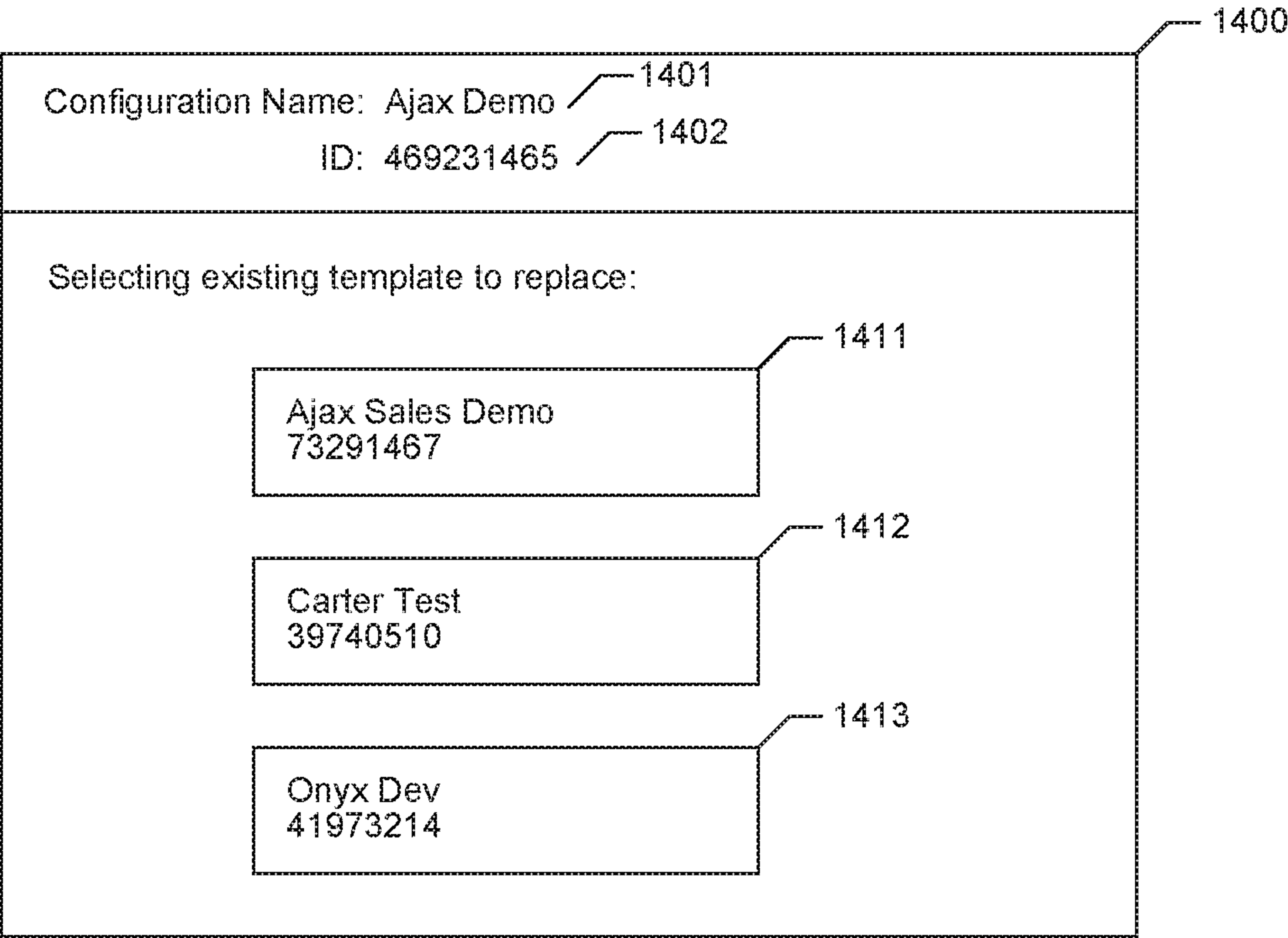


FIG. 14

MULTI-REGION VIRTUAL DATA CENTER TEMPLATE

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of priority to U.S. Provisional Patent Application No. 62/078,367 entitled “MULTI-REGION VIRTUAL DATA CENTER TEMPLATE,” filed on Nov. 11, 2014, which is hereby incorporated by reference in its entirety.

BACKGROUND

[0002] Some virtual computing embodiments make use of environments. An environment is a virtual workspace, or “virtual data center,” that contains virtual infrastructure such as one or more virtual machines (VM), one or more networks, network connections, and one or more computational services. An environment also defines how VMs reach the public Internet, corporate data centers via VPN, and/or computational services external to the environment. Environments are used to contain and execute software applications. A typical example is a hotel reservation system. An environment may be running, suspended, stopped, or powered off. The exact state of an environment can be saved as a template. A template is a read-only definition of a complete virtual data center and includes the virtual infrastructure, computational services, applications state, and connections to external resources. Templates may be simple, containing only a single VM, or quite complex, containing hundreds of VMs spanning multiple networks. An environment can be instantiated from a template. Instantiating an environment from a template typically involves selecting one of a number of different physical data centers in which the template is to be instantiated as an environment. For instantiation in a remote data center selection, data constituting the template is copied from wherever it resides to the selected physical data center as part of the instantiation process.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] FIG. 1 is a network diagram showing an environment in which the facility operates in some embodiments.

[0004] FIG. 2 is a block diagram showing some of the components typically incorporated in at least some of the clients, servers, and other devices on which the facility operates.

[0005] FIG. 3 is a display diagram showing a typical visual user interface provided by the facility in some embodiments to permit a user to select from among available stored templates.

[0006] FIG. 4 is a display diagram showing a typical visual user interface presented by the facility in order to permit the user to select an action to perform with respect to a selected template.

[0007] FIG. 5 is a flow diagram showing steps typically performed by the facility in some embodiments in order to adjust the distribution of a template across data centers.

[0008] FIG. 6 is a display diagram showing a typical visual user interface presented by the facility in some embodiments to permit the user to adjust the geographic replication of a selected template.

[0009] FIG. 7A is a table diagram showing sample contents of a templates table used by the facility in some embodiments to track templates created from environments.

[0010] FIG. 7B is a table diagram showing sample contents of a template instances table used by the facility in some embodiments to track the geographic replication of instances of each template.

[0011] FIG. 8 is a storage diagram showing sample contents of persistent storage devices such as hard disks at each of a number of different physical data centers that can store copies of a template.

[0012] FIG. 9 is a flow diagram showing steps typically performed by the facility in some embodiments in order to instantiate an environment from a selected template.

[0013] FIG. 10 is a display diagram showing a typical visual user interface presented by the facility in some embodiments in order to present a list of available data centers.

[0014] FIG. 11 is an instantiation diagram depicting the instantiation of an environment by the facility.

[0015] FIG. 12 is a flow diagram showing steps typically performed by the facility in order to create a template from an environment to represent the state of the environment at the time the template is created.

[0016] FIG. 13 is a display diagram showing a typical visual user interface presented by the facility in some embodiments to receive a request to create a template from an environment.

[0017] FIG. 14 is a display diagram showing a typical visual user interface presented by the facility in order to enable the user to select an existing template to replace with a new template.

DETAILED DESCRIPTION

[0018] The inventors have recognized that conventional processes for instantiating an environment from a template suffer from unwelcome latency that results from the time required to copy the data constituting the template across significant geographic distances to the physical data center where the environment is instantiated. As the number of templates and desired locations increase, these problems become more acute.

[0019] Accordingly, the inventors have conceived and reduced to practice a software and/or hardware facility (“the facility”) that maintains the data making up a template simultaneously at multiple selected physical data centers. This allows an environment to be quickly instantiated from a single template at any of these physical data centers, without having to incur the time cost of copying the template data at the time of environment instantiation.

[0020] In some embodiments, the facility provides a user interface that a user may use to specify for the template, from among a list of available physical data centers, those in which instances of the template should be immediately available to quickly instantiate the environment represented by the template. These copies at each of multiple data centers are referred to herein as “instances” of the template.

[0021] In some embodiments, the template remains monolithic, in that a single template reference refers to the template and all of its distributed instances. In such embodiments, the template and all of its distributed instances can be deleted or updated with a single operation performed with respect to the template.

[0022] In some embodiments, the facility enables a user to replace an existing template with a new template based upon the state of an environment. After doing so, references that formerly pointed to the original template point to the new

template. In some embodiments, the facility provides a user interface that the user may use in order to identify the existing template that is to be saved over, and the environment from whose state the new template is to be generated (i.e., whose state is to be reflected by the replacement template).

[0023] In some embodiments, in the facility's replication of a template instance between data centers, the facility identifies in the template instance being replicated one or more region-specific resources; these are resources that are usable in the context of one or more geographic regions, but not usable in others. For example, in some embodiments, an environment, and correspondingly a template created from that environment and its instances, can make a particular virtual machine of the environment accessible via the Internet at a particular IP address. Because Internet-valid IP addresses are assigned in distinct blocks by regional authorities, an external IP address assigned to a virtual machine that is part of an environment instantiated at a first center is often not usable to connect to the corresponding virtual machine at a different data center. Some additional types of resources are region-specific based on their reliance on an external IP address, including resources such as VPN tunnels to connect a virtual machine, via its external IP address, securely to an external network node; and instances Inter-Environment Network Routing to connect a virtual machine, via its external IP address, to a virtual machine of a different environment. When the facility identifies one or more such regional-specific resources in a template instance being replicated to a different data center, in the new template instance, the facility substitutes for each region-specific resource a corresponding region-specific resource that can be used in the target data center for the for the new template instance. In various embodiments, in performing this substitution, the facility selects from a pool of external IP addresses or region-specific resources of other types available at the target data center. In some embodiments, this is done automatically by the facility; in some embodiments, this is done by the facility based on user input to select the IP addresses.

[0024] By performing in some or all of the manners described above, the facility makes the use of templates in a virtual computing environment-such as a distributed virtual computing environment-more useful.

[0025] FIG. 1 is a network diagram showing an environment in which the facility operates in some embodiments. The environment includes a client computer system **110**. This client is used by a user to interact with virtual computing environments hosted in any of a number of physical data centers, such as physical data centers **131-133**. These physical data centers can be located in a variety of different geographic locations. The client interacts with each physical data center hosting virtual computing environments via the Internet **120** or another network. In various embodiments, this communication enables the user to instantiate an environment-such as from a template; monitor and control execution of the environment; and save the state of the environment as a template. In some embodiments, the user performs these interactions using a web browser program executing on the client computer system, which interacts with a web server located in each of the physical data centers. In some embodiments, the facility exposes an Application Programming Interface to allow template instances to be managed programmatically.

[0026] FIG. 2 is a block diagram showing some of the components typically incorporated in at least some of the clients, servers, and other devices on which the facility operates. In various embodiments, these devices **200** can include server computer systems, desktop computer systems, laptop computer systems, netbooks, mobile phones, personal digital assistants, televisions, cameras, automobile computers, electronic media players, computer cards, etc. In various embodiments, the computer systems and devices include zero or more of each of the following: a central processing unit ("CPU") **201** for executing computer programs; a computer memory **202** for storing programs and data while they are being used, including the facility and associated data, an operating system including a kernel, and device drivers; a persistent storage device **203**, such as a hard drive or flash drive for persistently storing programs and data; a computer-readable media drive **204**, such as a floppy, CD-ROM, or DVD drive, for reading programs and data stored on a computer-readable medium; and a network connection **205** for connecting the computer system to other computer systems to send and/or receive data, such as via the Internet or another network and its networking hardware, such as switches, routers, repeaters, electrical cables and optical fibers, light emitters and receivers, radio transmitters and receivers, and the like. While computer systems configured as described above are typically used to support the operation of the facility, those skilled in the art will appreciate that the facility may be implemented using devices of various types and environments, and having various components.

[0027] FIG. 3 is a display diagram showing a typical visual user interface provided by the facility in some embodiments to permit a user to select from among available stored templates. A display **300** includes direction to select a template, and controls **311-313** each for selecting a different stored template. For example, the user may click on or otherwise activate control **311** to select an "Ajax Sales Demo" template having the template identifier "73291467."

[0028] FIG. 4 is a display diagram showing a typical visual user interface presented by the facility in order to permit the user to select an action to perform with respect to a selected template. The display **400** includes the name **401** and the identifier **402** of the template selected by activating control **311** in FIG. 3. The display further contains controls **411-413**. The user can activate control **411** in order to adjust the distribution of instances of the selected template among the available data centers. The user can activate control **412** in order to instantiate a new environment using the selected template. The user can activate control **413** in order to delete the template and all associated template instances.

[0029] FIG. 5 is a flow diagram showing steps typically performed by the facility in some embodiments in order to adjust the replication across data centers. For example, in some embodiments, the facility performs the steps shown in FIG. 5 in response to the user activating control **411** shown in FIG. 4. In step **501**, the facility presents a list of available data centers and identifies those data centers presently selected to store instances of the selected template. In step **502**, the facility receives from the user input specifying an updated set of data center selections. For example, the user can select formerly unselected data centers, and deselect formerly selected data centers. In step **503**, for each newly selected data center reflected in the user input received in step **502**, the facility replicates template instances to that

data center and causes the template to refer to them. In some embodiments, the facility performs mapping of region-specific resources in the replicated template instances to be usable in each target data center, such as in some or all of the manners described above. In step **504**, for each newly deselected data center specified by the user input received in step **502**, the facility deletes the template instances from this data center and removes the reference from the template to the deleted template instances. After step **504**, these steps conclude.

[0030] Those skilled in the art will appreciate that the steps shown in FIG. **5** and in each of the flow diagrams discussed below may be altered in a variety of ways. For example, the order of the steps may be rearranged; some steps may be performed in parallel; shown steps may be omitted, or other steps may be included; a shown step may be divided into substeps, or multiple shown steps may be combined into a single step, etc.

[0031] FIG. **6** is a display diagram showing a typical visual user interface presented by the facility in some embodiments to permit the user to adjust the geographic replication of a selected template. The display **600** contains the selected template's name **601** and identifier **602**. It further directs the user to select the data centers among data centers **611-618** to which instances of the template should be replicated. It can be seen here that the user has selected data centers **611**, **612**, and **614** by checking the associated boxes. The display also contains a control **621** that the user can activate in order to select all of the listed data centers, as well as a control **622** that the user can select in order to deselect all of the listed data centers, effectively deleting the template and all of its instances. Based upon the user input provided as shown in FIG. **6**, the facility will replicate template instances for the selected template to any of data centers **611**, **612**, and **614** where it is not presently stored, and delete template instances for the selected template from any of data centers **613** and **615-618** where they are presently stored.

[0032] FIG. **7A** is a table diagram showing sample contents of a templates table used by the facility in some embodiments to track templates created from environments. The templates table **700** is made up of rows each corresponding to a different template, such as row **701**. Each row is divided into the following columns: a template identifier column **711** containing an identifier uniquely identifying the template; and a template name column **712** containing a name specified for the template to which the row corresponds. The facility uses each row's template identifier to connect it to instances of the template stored at particular data centers.

[0033] While FIG. **7A** and the table diagrams discussed below show a table whose contents and organization are designed to make them more comprehensible by a human reader, those skilled in the art will appreciate that actual data structures used by the facility to store this information may differ from the table shown, in that they, for example, may be organized in a different manner; may contain more or less information in each row than shown; may be compressed and/or encrypted; may contain a much larger number of rows than shown, etc.

[0034] FIG. **7B** is a table diagram showing sample contents of a template instances table used by the facility in some embodiments to track the geographic replication of instances of each template. The template instances table **700** is made up of rows each corresponding to a different

template instance, such as rows **751-753**. Each row is divided into the following columns: a template instance identifier column **761** containing an identifier identifying the template instance to which the row corresponds; a template id column **762** containing a template identifier identifying the template of which the template instance to which the row corresponds is an instance; a location column **763** identifying the data center in which the template instance is stored; a creation time column **764** indicating the date and time at which the template instance was stored at the data center; and a pointer to saved state column **765** containing information usable to access the template instanced stored at the data center, such as a network path or other file identifier. It can be seen from the contents of rows **701** in FIG. **7A** and row **751** in FIG. **7B** that an instance of the "Ajax Sales Demo" template having template identifier 73291467 was stored at the Beijing data center on Jan. 15, 2014 at 14:07:03.211 and can be accessed there using <pointer1>. It can be seen from rows **752-753** in FIG. **7B** that additional instances of this template are also stored in the Sydney and Chicago data centers. Each time a user changes the geographic distribution of a template such as by using the visual user interface shown in FIG. **6**, the facility updates the template instances table to reflect these changes. Additionally, each time a new template is created, the facility adds a row corresponding to the created template to the template table, and at least one new template instance to the template instances table for the new template. In some embodiments (not shown), the template instances table further includes additional columns, such as a version column identifying the version of a template instance to which each row corresponds, and a superseded column indicating whether the template instance to which the row corresponds has been superseded by a newer version, or has been deleted.

[0035] FIG. **8** is a storage diagram showing sample contents of persistent storage devices such as hard disks at each of a number of different physical data centers. These include one or more disks **810** at the Beijing data center, one or more disks **820** at the Sydney data center, one or more disks **830** at the Alexandria data center, one or more disks **840** at the Chicago data center, one or more disks **850** at the San Francisco data center, one or more disks **860** at the Tokyo data center, one or more disks **870** at the Berlin data center, and one or more disks **880** at the Dubai data center. It can be seen that, consistent with the selection of the Beijing, Sydney, and Chicago data centers reflected in FIGS. **6** and **7**, the facility has stored an instance **92837462** of the **73291467** template on the Beijing disks **810** (as template instance **811**), instance **44318873** on the Sydney disks **820** (as template instance **821**), and instance **11062344** on the Chicago disks **840** (as template instance **841**). For any newly-selected data center, the facility will copy template instances to that data center's disks. For any newly-deselected data center, the facility will delete template instances from that data center's disks. If the template is replaced with a new template, the facility will replace each template instance of the original template on a data centers' disks with a copy of the replacement template instance create from the contents of the selected environment. If the template is deleted, the facility will remove the template and all template instances from each of the data centers' disks.

[0036] FIG. **9** is a flow diagram showing steps typically performed by the facility in some embodiments in order to instantiate an environment from a selected template. For

example, in some embodiments, the facility performs these steps when the user activates control **412** shown in FIG. **4**. In step **901**, the facility presents a list of data centers.

[0037] FIG. **10** is a display diagram showing a typical visual user interface presented by the facility in some embodiments in order to present a list of available data centers. The display **1000** includes the name **1001** and identifier **1002** of the selected template. The display further includes a direction to select a data center in which to instantiate an environment from the selected template. Further included are controls **1011-1018**, each corresponding to a different data center and available for the user to activate in order to instantiate an environment from the selected template in that data center. It can be seen that controls **1011**, **1012**, and **1014**, corresponding to the Beijing, Sydney, and Chicago data centers, respectively, contain an extra indication that instantiation of the environment will proceed quickly in any of those data centers as the result of the selected template having instances stored at those data centers.

[0038] For example, the user may operate control **1011** in order to cause an environment to be instantiated from the selected template in the Beijing data center.

[0039] Returning to FIG. **9**, in step **902**, the facility receives user input selecting a data center. In step **903**, if the template is stored at the selected data center, then the facility continues to step **905**, else the facility continues in step **904**. In some embodiments, the facility determines whether a template instance is stored at the selected data center in step **903** by consulting the template instance table shown in FIG. **7B**, such as at the intersection of row **751** and column **763** for the combination of the Ajax Sales Demo template in the Beijing data center. When an instance is stored in the selected data center, the flow proceeds directly to step **905** and quickly instantiates an environment in the selected data center from the local template instance data. In step **904**, when a template instance is not stored in the selected data center, the facility copies the data from another remote template instance to the selected data center to instantiate an environment. In some embodiments, before performing step **904**, the facility confirms with the user that it should do so. After step **904**, the facility continues in step **905**. In step **905**, the facility uses the copied data to instantiate an environment at the selected data center. After step **905**, these steps conclude.

[0040] FIG. **11** is an instantiation diagram depicting the instantiation of an environment by the facility. It can be seen that the template instance **1111** stored on the disks **1110** at the Beijing data center is used to instantiate environment **1121**. The instantiation clones the stored data in the template instance to create environment **1112** located at the Beijing data center. Because the resulting environment is located within the same data center as the template instance, and in some embodiments on the same disks, the instantiation operation completes faster than cross-data center environment instantiation.

[0041] FIG. **12** is a flow diagram showing steps typically performed by the facility to create a template from an environment in order to represent the state of the environment at the time the template is created. Such a template can be used to replicate a large number of copies of the environment, to later return to this earlier state of the environment for either assessment or execution, etc.

[0042] FIG. **13** is a display diagram showing a typical visual user interface presented by the facility in some embodiments to receive a request to create a template from an environment. The display **1300** contains the name **1301** and the identifier **1302** of a selected environment. It further includes consoles **1311** and **1312** each showing a visual representation of a different one of the virtual machines included in the environment. The display further includes a control **1321** that the user can activate in order to pause and resume execution of the virtual machines making up the environment. The display further includes a control **1322** that the user can activate in order to save the current state of the environment as a template.

[0043] Returning to FIG. **12**, in step **1202**, after receiving a request to create a template from a selected environment, such as by activation of control **1322** in FIG. **13**, the facility prompts the user to specify whether the template in which the state of an environment is to be saved will be a new template, or whether it will be stored in place of an existing template. In step **1203**, if the user specifies that it will be a new template, then the facility continues in step **1204**, whereas if the user specifies that it will replace an existing template, the facility continues in step **1206**. In step **1204**, the facility creates a template and adds it to the templates table. In step **1205**, the facility creates a new template instance from the environment and adds it to the template instance table with a pointer to the template created in **1204**. In some embodiments (not shown), the facility automatically transitions to the process shown in FIG. **5** and the visual user interface shown in FIG. **6** to allow the user to specify the data centers in which the newly created template is replicated. After step **1205**, these steps conclude.

[0044] In step **1206**, where the user has specified to store the environment to be saved in place of an existing template, the facility displays a list of the existing templates.

[0045] FIG. **14** is a display diagram showing a typical visual user interface presented by the facility in order to enable the user to select an existing template to update with the current state of the selected environment. The display **1400** includes the name **1401** and identifier **1402** of the environment whose state will be used to create the template instance that will replace the template instances of the selected template. It further includes the direction to select an existing template whose instances are to be replaced, and controls **1411-1413** each corresponding to an existing template that is available to be updated by the current state of the selected environment. For example, the user can activate control **1411** in order to update the Ajax Sales Demo template by replacing the existing template instances with new template instances containing the current state of the selected "Ajax demo" environment.

[0046] Returning to FIG. **12**, in step **1207**, the facility receives user input selecting an existing template for update. In step **1208**, in a manner similar to that used in step **1205**, the facility creates a new template instance from the selected environment and adds it to the template instance table with a pointer to the selected existing template. In step **1209**, the facility stores the created template in place of an existing template instance of the template selected by the user input received in step **1207**. When template instances are stored in multiple locations, in step **1210**, the facility replicates template instances to all data centers having an instance of that template. In some embodiments, steps **1208-1210** appear to be atomic from the perspectives of end users. In the case of

the Ajax Sales Demo template selected as the template to update, the facility creates a new template instance from the current state of the selected environments. The template instance is replicated to all data center disks Beijing disks **810**, Sydney disks **820**, and Chicago disks **840** containing the old template instances. The old template instances are then removed from the template instance table. In some embodiments (not shown), the facility does not remove old template instance from the template instances table shown in FIG. 7B. In various embodiments, this allows the template to refer to both old and new versions of a template. After step **1210**, these steps conclude.

[0047] In the figures discussed above, data centers in which templates may be stored and environments instantiated are identified based upon textual names for the cities in which they are located or near which they are located. In various embodiments, the facility identifies data centers in a variety of other ways, such as counties, states, countries, continents, or other types of geographic regions in which they are located. In various embodiments, the data centers are identified by showing their locations on a map, or by showing on a map regions containing the data centers.

[0048] It will be appreciated by those skilled in the art that the above-described facility may be straightforwardly adapted or extended in various ways. While the foregoing description makes reference to particular embodiments, the scope of the invention is defined solely by the claims that follow and the elements recited therein.

1-26. (canceled)

27. A method in a computing system for managing the geographic distribution of instances of a template recording a past state of a virtual computing environment, the method comprising:

receiving user input specifying information uniquely identifying a template, among a plurality of templates, to be deleted; and

in response to the receiving:

identifying a plurality of physical data centers in which a template instance of the specified template is stored, wherein the template instance is a copy of the specified template that was previously created in that physical data center;

deleting from each of the identified physical data centers the stored template instance; and

updating a data structure storing an association between the specified template and the template instance by removing the association.

28. A hardware networking component conveying a user interface data structure, the data structure comprising:

information directing the presentation of a visual indication of a template recording a past state of a virtual computing environment;

information directing the presentation of visual indications of each of a plurality of geographic locations; and

information directing the presentation of one or more controls usable to select any among the presented visual indications of each of a plurality of geographic locations as geographic locations in which the indicated template is to be maintained, such that, upon receipt, the information comprising the data structure is usable to facilitate specification by a user of geographic locations at which the indicated template is to be maintained.

29. A method in a computing system for recording a current state of a virtual computing environment, the method comprising:

presenting a list of existing templates each stored persistently;

receiving user input selecting one of the listed existing templates, the selected existing template referring to a first template instance;

in response to the receiving,

persistently storing a second template instance capturing the current state of the virtual computing environment; and

causing the selected template to refer to the second template instance rather than the first template instance.

30. The method of claim **29** wherein, at the time the user input is received, the selected existing template referring to both a first template instance at a first data center and a third template instance at a second data center,

and wherein persistently storing the second template instance comprises persistently storing the second template instance at the first data center,

the method further comprising:

persistently storing a fourth template instance at the second data center, and

causing the selected template to refer to the fourth template instance as well as the second template instance.

31. The method of claim **30**, further comprising formulating the fourth template instance by:

copying the second template instance; and

in the copy, substituting for each of one or more region-specific resources suited to the first data center a corresponding region-specific resource suited to the second data center.

32. The method of claim **29**, further comprising:

at a time before the storing, receiving a first request to dereference a reference to the existing template selected by the received user input;

in response to the first request, identifying the first template instance;

at a time after the storing, receiving a second request to dereference the reference; and

in response to the second request, identifying the second template instance.

33. One or more computer memories collectively containing a template instances table data structure, the data structure comprising:

at least one entry, each entry comprising:

information identifying a template recording a particular past state of a virtual computing environment, the template comprising a state for each of a plurality of virtual machines specified for the virtual computing environment; and

information identifying a physical data center in which an instance of data constituting the identified template has been replicated.

34. The computer memories of claim **33** wherein at least two of the entries contain information identifying the same template.

35. The computer memories of claim **33** wherein each entry of the data structure further comprises:

information usable to access in the identified physical data center the instance of data constituting the identified template that has been replicated in the identified physical data center.

36. The computer memories of claim **33** wherein each entry of the data structure further comprises:

information identifying a version of the data constituting the identified template.

37. A hardware networking component conveying a user interface data structure, the data structure comprising:

information directing the presentation of a list of existing templates each stored persistently; and

information directing the presentation of one or more controls usable to select one of the listed existing templates as an existing template that is to be replaced by a new template recording a current state of a virtual computing environment, such that, upon receipt, the information comprising the data structure is usable to facilitate specification by a user of an existing template that is to be replaced by a new template recording a current state of the virtual computing environment.

38. One or more computer memories collectively storing contents configured to cause a computing system to perform a method for recording a current state of a virtual computing environment, the method comprising:

accessing an indication of an existing template stored persistently; and

persistently storing a distinguished template capturing the current state of the virtual computing environment in place of the indicated existing template.

39. One or more computer memories collectively containing a reference data structure, the data structure comprising:

a reference to a persistently stored first template recording a first state of a first virtual computing environment,

such that, up to a time when the first template is replaced with a second template recording a second state of a second computing environment, the reference refers to the first template and can be used to instantiate a new virtual computing environment based on the first template,

and such that, after the time when the first template is replaced with the second template, the reference refers to the second template and can be used to instantiate a new virtual computing environment based on the second template.

40. The computer memories of claim **39** wherein the second virtual computing environment is distinct from the first virtual computing environment.

41. The computer memories of claim **39** wherein the reference is a numerical identifier.

42. The computer memories of claim **39** wherein the reference is a textual identifier.

43. The computer memories of claim **39** wherein the reference is a URL.

44. The computer memories of claim **39** wherein the reference identifies a particular location in a particular memory at which the template is stored.

45. The computer memories of claim **39** wherein the reference directly identifies a particular location in a particular memory at which the template is stored.

46. The computer memories of claim **39** wherein the reference indirectly identifies a particular location in a particular memory at which the template is stored.

* * * * *