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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 August 2023 (10.08.2023)

WIPO | PCT

(10) International Publication Number
WO 2023/150531 A1

(51) International Patent Classification:
*G06F 9/50* (2006.01) *G06F 21/30* (2013.01)

(21) International Application Number:
PCT/US2023/061723

(22) International Filing Date:
01 February 2023 (01.02.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

| | | |
|---|---|---|
| 63/306,007 | 02 February 2022 (02.02.2022) | US |
| 63/306,918 | 04 February 2022 (04.02.2022) | US |
| 63/321,614 | 18 March 2022 (18.03.2022) | US |
| 63/333,965 | 22 April 2022 (22.04.2022) | US |
| 63/336,701 | 29 April 2022 (29.04.2022) | US |
| 63/336,811 | 29 April 2022 (29.04.2022) | US |
| 63/339,297 | 06 May 2022 (06.05.2022) | US |
| 63/357,541 | 30 June 2022 (30.06.2022) | US |
| 63/388,921 | 13 July 2022 (13.07.2022) | US |

(71) Applicant: **ORACLE INTERNATIONAL CORPORATION** [US/US]; 500 Oracle Parkway, M/S 5OP7, Redwood Shores, California 94065 (US).

(72) Inventors: **NAGARAJA, Girish**; c/o Oracle International Corporation, 500 Oracle Parkway, M/S 50P7, Redwood Shores, California 94065 (US). **EVANI, Venkata Subbarao**; c/o Oracle International Corporation, 500 Oracle Parkway, M/S 50P7, Redwood Shores, California 94065 (US). **VOGEL, Daniel, M.**; c/o Oracle International Corporation, 500 Oracle Parkway, M/S 50P7, Redwood Shores, California 94065 (US). **RUSHTON, Matthew, Victor**; c/o Oracle International Corporation, 500 Oracle Parkway, M/S 50P7, Redwood Shores, California 94065 (US).

(54) Title: CLOUD-LINK ADAPTOR OF A MULTI-CLOUD INFRASTRUCTURE

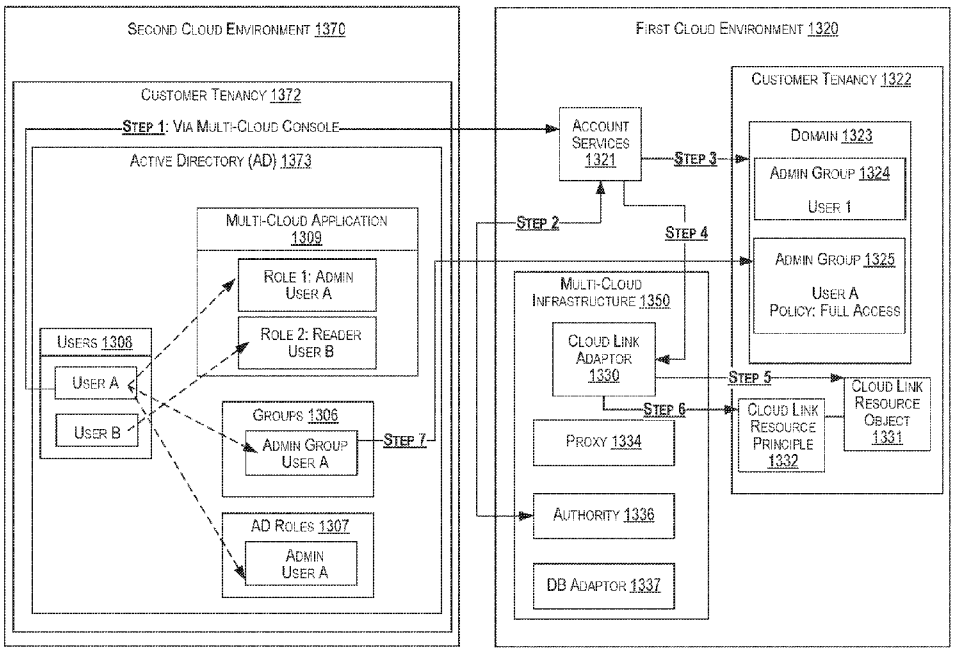


FIG. 13

(57) Abstract: Techniques are described for providing a multi-cloud control plane (MCCP) in a first cloud infrastructure (included in a first cloud environment provided by a first cloud services provider) that enables services and/or resources provided in the first cloud infrastructure to be utilized by users of a second cloud environment. The first cloud infrastructure receives a request from a user associated with an account in the second cloud infrastructure. The request corresponding to using a service provided by the first cloud infrastructure. A tenancy is created for the user in the first cloud infrastructure to enable the user to utilize the service, and a link-resource object is created that includes information linking the tenancy of the user in the first cloud infrastructure to the account of the user in the second cloud infrastructure, the link-resource object enabling the user to utilize the service provided by the first cloud infrastructure.

WO 2023/150531 A1

[Continued on next page]

(74) **Agent: GADKAR, Arush** et al.; 1100 Peachtree Street, Suite 2800, Mailstop: IP Docketing - 22, Atlanta, Georgia 30309 (US).

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

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

**Published:**

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

# CLOUD-LINK ADAPTOR OF A MULTI-CLOUD INFRASTRUCTURE

## CROSS-REFERENCES TO RELATED APPLICATIONS

**[0001]** The present application is a non-provisional of and claims the benefit of each of the following provisional applications. The entire contents of each of the provisional applications listed below are incorporated herein by reference for all purposes.

(1) U.S. Provisional Application No. 63/306,007, filed on February 2, 2022;

(2) U.S. Provisional Application No. 63/306,918, filed on February 4, 2022;

(3) U.S. Provisional Application No. 63/321,614, filed on March 18, 2022;

(4) U.S. Provisional Application No. 63/333,965, filed on April 22, 2022;

(5) U.S. Provisional Application No. 63/336,811, filed on April 29, 2022;

(6) U.S. Provisional Application No. 63/336,701, filed on April 29, 2022;

(7) U.S. Provisional Application No. 63/339,297, filed on May 6, 2022;

(8) U.S. Provisional Application No. 63/357,541, filed on June 30, 2022; and

(9) U.S. Provisional Application No. 63/388,921, filed on July 13, 2022.

## FIELD

**[0002]** The present disclosure relates to cloud architectures, and more particularly to techniques for linking two cloud environments such that a user of one cloud environment can use a service provided by the other cloud environment.

## BACKGROUND

**[0003]** The last few years have seen a dramatic increase in the adoption of cloud services and this trend is only going to increase. Various different cloud environments are being provided by different cloud service providers (CSPs), each cloud environment providing a set of one or more cloud services. The set of cloud services offered by a cloud environment may include one or more different types of services including but not restricted to Software-as-a-Service (SaaS) services, Infrastructure-as-a-Service (IaaS) services, Platform-as-a-Service (PaaS) services, and others.

[0004] While various different cloud environments are currently available, each cloud environment provides a closed ecosystem for its subscribing customers. As a result, a customer of a cloud environment is restricted to using the services offered by that cloud environment. There is no easy way for a customer subscribing to a cloud environment provided by a CSP to, via that cloud environment, use a service offered in a different cloud environment provided by a different CSP. Embodiments discussed herein address these and other issues. Embodiments discussed herein address these and other issues.

## SUMMARY

[0005] The present disclosure relates generally to improved cloud architectures, and more particularly to techniques for linking two clouds such that a user of one cloud environment can use a service provided by another different cloud environment. Various embodiments are described herein, including methods, systems, non-transitory computer-readable storage media storing programs, code, or instructions executable by one or more processors, and the like. Some embodiments may be implemented by using a computer program product, comprising computer program/instructions which, when executed by a processor, cause the processor to perform any of the methods described in the disclosure.

[0006] Embodiments of the present disclosure provide for a multi-cloud control plane (MCCP) framework that provisions for capabilities to deliver services of a particular cloud network (e.g., Oracle Cloud Infrastructure (OCI)) to users on other clouds (e.g., in Microsoft Azure). The MCCP framework allows users (of other cloud environment(s)) to access services (e.g., PaaS services) of a cloud environment, while providing with a user experience as close as possible to that of the native cloud environment(s) of the user. A key value proposition to MCCP is that customers will be able to experience the full data plane capabilities of the services in external clouds.

[0007] One embodiment of the present disclosure is directed to a method comprising: receiving, by a first cloud infrastructure provided by a first cloud services provider (CSP), a request from a user associated with an account in a second cloud infrastructure provided by a second CSP, the request requesting use of a service provided by the first cloud infrastructure; creating a tenancy for the user in the first cloud infrastructure to enable the user to utilize the service; and creating, by the first cloud infrastructure, a link-resource object that includes information linking the tenancy of the user in the first cloud infrastructure to the account of the

user in the second cloud infrastructure, the link-resource object enabling the user to utilize the service provided by the first cloud infrastructure.

**[0008]** An aspect of the present disclosure provides for a computing device comprising one or more data processors, and a non-transitory computer readable storage medium containing instructions which, when executed on the one or more data processors, cause the computing device to perform part or all of one or more methods disclosed herein.

**[0009]** Another aspect of the present disclosure provides for a computer-program product tangibly embodied in a non-transitory machine-readable storage medium, including instructions configured to cause one or more data processors to perform part or all of one or more methods disclosed herein.

**[0010]** The foregoing, together with other features and embodiments will become more apparent upon referring to the following specification, claims, and accompanying drawings.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0011]** Features, embodiments, and advantages of the present disclosure are better understood when the following Detailed Description is read with reference to the accompanying drawings.

**[0012]** FIG. 1 is a high-level diagram of a distributed environment showing a virtual or overlay cloud network hosted by a cloud service provider infrastructure according to certain embodiments.

**[0013]** FIG. 2 depicts a simplified architectural diagram of the physical components in the physical network within CSPI according to certain embodiments.

**[0014]** FIG. 3 shows an example arrangement within CSPI where a host machine is connected to multiple network virtualization devices (NVDs) according to certain embodiments.

**[0015]** FIG. 4 depicts connectivity between a host machine and an NVD for providing I/O virtualization for supporting multitenancy according to certain embodiments.

**[0016]** FIG. 5 depicts a simplified block diagram of a physical network provided by a CSPI according to certain embodiments.

**[0017]** FIG. 6 depicts a simplified high-level diagram of a distributed environment comprising multiple cloud environments provided by different cloud service providers (CSPs) wherein the cloud environments include a particular cloud environment that provides specialized infrastructure that enables one or more cloud services provided by that particular

cloud environment to be used by customers of other cloud environments according to certain embodiments.

**[0018]** FIG. 7 depicts an exemplary high-level architecture of a multi-cloud control plane (MCCP), according to some embodiments.

**[0019]** FIGs. 8A and 8B depict exemplary processes for linking two user accounts in different cloud environments, according to some embodiments.

**[0020]** FIG. 9 depicts an exemplary system diagram illustrating components of a multi-cloud control plane (MCCP) according to some embodiments.

**[0021]** FIG. 10 depicts an exemplary request received by a multi-cloud infrastructure, according to some embodiments.

**[0022]** FIG. 11 illustrates an exemplary flowchart depicting steps performed by a multi-cloud infrastructure, according to certain embodiments.

**[0023]** FIG. 12 illustrates an exemplary relationship diagram of a link-resource object, according to certain embodiments.

**[0024]** FIG. 13 depicts a flow diagram illustrating an example process of linking two user accounts in different cloud environments, according to certain embodiments.

**[0025]** FIG. 14 illustrates a flow diagram illustrating an example process of using a cloud-link resource object in deploying resources, according to certain embodiments.

**[0026]** FIG. 15 depicts a flow diagram illustrating steps performed in a token exchange process, according to certain embodiments.

**[0027]** FIG. 16 depicts a schematic illustrating an exemplary token exchange process, in accordance with some embodiments.

**[0028]** FIG. 17 depicts a flowchart illustrating a process performed by a multi-cloud infrastructure, according to certain embodiments.

**[0029]** FIG. 18 is a block diagram illustrating one pattern for implementing a cloud infrastructure as a service system, according to at least one embodiment.

**[0030]** FIG. 19 is a block diagram illustrating another pattern for implementing a cloud infrastructure as a service system, according to at least one embodiment.

**[0031]** FIG. 20 is a block diagram illustrating another pattern for implementing a cloud infrastructure as a service system, according to at least one embodiment.

**[0032]** FIG. 21 is a block diagram illustrating another pattern for implementing a cloud infrastructure as a service system, according to at least one embodiment.

[0033] FIG. 22 is a block diagram illustrating an example computer system, according to at least one embodiment.

## DETAILED DESCRIPTION

[0034] In the following description, for the purposes of explanation, specific details are set forth in order to provide a thorough understanding of certain embodiments. However, it will be apparent that various embodiments may be practiced without these specific details. The figures and description are not intended to be restrictive. The word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment or design described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments or designs.

[0035] The present disclosure relates generally to improved cloud architectures, and more particularly to techniques for linking two clouds such that a user of one cloud environment can use a service provided by another different cloud environment. Various embodiments are described herein, including methods, systems, non-transitory computer-readable storage media storing programs, code, or instructions executable by one or more processors, and the like. Some embodiments may be implemented by using a computer program product, comprising computer program/instructions which, when executed by a processor, cause the processor to perform any of the methods described in the disclosure.

[0036] Embodiments of the present disclosure provide for a multi-cloud control plane (MCCP) framework that provisions for capabilities to deliver services of a particular cloud network (e.g., Oracle Cloud Infrastructure (OCI)) to users on other clouds (e.g., in Microsoft Azure). The MCCP framework allows users (of other cloud environment(s)) to access services (e.g., PaaS services) of a cloud environment, while providing with a user experience as close as possible to that of the native cloud environment(s) of the user. A key value proposition to MCCP is that customers will be able to experience the full data plane capabilities of the services in external clouds.

[0037] The MCCP enables users of a second cloud infrastructure (e.g., Azure users) to make use of resources (e.g., database resources) provided by a first cloud infrastructure (e.g., OCI) in a way that is transparent to the user. Specifically, the services provided by the first cloud infrastructure appear as "native" services in the second cloud infrastructure. This allows customers of the second cloud infrastructure to natively access services provided by the first cloud infrastructure. As will be described below with reference to FIGs. 6-17, the MCCP is a

collection of microservices executed in the first cloud infrastructure which exposes resources of the first cloud infrastructure to be utilized by external cloud users (e.g., users of the second cloud infrastructure). Each of the microservices acts as a proxy providing communication to resources provided by the first cloud infrastructure.

**Examples of Cloud Networks**

**[0038]** The term cloud service is generally used to refer to a service that is made available by a cloud services provider (CSP) to users or customers on demand (e.g., via a subscription model) using systems and infrastructure (cloud infrastructure) provided by the CSP. Typically, the servers and systems that make up the CSP's infrastructure are separate from the customer's own on-premises servers and systems. Customers can thus avail themselves of cloud services provided by the CSP without having to purchase separate hardware and software resources for the services. Cloud services are designed to provide a subscribing customer easy, scalable access to applications and computing resources without the customer having to invest in procuring the infrastructure that is used for providing the services.

**[0039]** There are several cloud service providers that offer various types of cloud services. There are various different types or models of cloud services including Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS), and others.

**[0040]** A customer can subscribe to one or more cloud services provided by a CSP. The customer can be any entity such as an individual, an organization, an enterprise, and the like. When a customer subscribes to or registers for a service provided by a CSP, a tenancy or an account is created for that customer. The customer can then, via this account, access the subscribed-to one or more cloud resources associated with the account.

**[0041]** As noted above, infrastructure as a service (IaaS) is one particular type of cloud computing service. In an IaaS model, the CSP provides infrastructure (referred to as cloud services provider infrastructure or CSPI) that can be used by customers to build their own customizable networks and deploy customer resources. The customer's resources and networks are thus hosted in a distributed environment by infrastructure provided by a CSP. This is different from traditional computing, where the customer's resources and networks are hosted by infrastructure provided by the customer.

**[0042]** The CSPI may comprise interconnected high-performance compute resources including various host machines, memory resources, and network resources that form a physical network, which is also referred to as a substrate network or an underlay network. The resources in CSPI may be spread across one or more data centers that may be geographically

spread across one or more geographical regions. Virtualization software may be executed by these physical resources to provide a virtualized distributed environment. The virtualization creates an overlay network (also known as a software-based network, a software-defined network, or a virtual network) over the physical network. The CSPI physical network provides the underlying basis for creating one or more overlay or virtual networks on top of the physical network. The physical network (or substrate network or underlay network) comprises physical network devices such as physical switches, routers, computers and host machines, and the like. An overlay network is a logical (or virtual) network that runs on top of a physical substrate network. A given physical network can support one or multiple overlay networks. Overlay networks typically use encapsulation techniques to differentiate between traffic belonging to different overlay networks. A virtual or overlay network is also referred to as a virtual cloud network (VCN). The virtual networks are implemented using software virtualization technologies (e.g., hypervisors, virtualization functions implemented by network virtualization devices (NVDs) (e.g., smartNICs), top-of-rack (TOR) switches, smart TORs that implement one or more functions performed by an NVD, and other mechanisms) to create layers of network abstraction that can be run on top of the physical network. Virtual networks can take on many forms, including peer-to-peer networks, IP networks, and others. Virtual networks are typically either Layer-3 IP networks or Layer-2 VLANs. This method of virtual or overlay networking is often referred to as virtual or overlay Layer-3 networking. Examples of protocols developed for virtual networks include IP-in-IP (or Generic Routing Encapsulation (GRE)) Virtual Extensible LAN (VXLAN — IETF RFC 7348), Virtual Private Networks (VPNs) (e.g., MPLS Layer-3 Virtual Private Networks (RFC 4364)), VMware's NSX, GENEVE (Generic Network Virtualization Encapsulation), and others.

**[0043]** For IaaS, the infrastructure (CSPI) provided by a CSP can be configured to provide virtualized computing resources over a public network (e.g., the Internet). In an IaaS model, a cloud computing services provider can host the infrastructure components (e.g., servers, storage devices, network nodes (e.g., hardware), deployment software, platform virtualization (e.g., a hypervisor layer), or the like). In some cases, an IaaS provider may also supply a variety of services to accompany those infrastructure components (e.g., billing, monitoring, logging, security, load balancing and clustering, etc.). Thus, as these services may be policy-driven, IaaS users may be able to implement policies to drive load balancing to maintain application availability and performance. CSPI provides infrastructure and a set of complementary cloud services that enable customers to build and run a wide range of applications and services in a

highly available hosted distributed environment. CSPI offers high-performance compute resources and capabilities and storage capacity in a flexible virtual network that is securely accessible from various networked locations such as from a customer's on-premises network. When a customer subscribes to or registers for an IaaS service provided by a CSP, the tenancy created for that customer is a secure and isolated partition within the CSPI where the customer can create, organize, and administer their cloud resources.

**[0044]** Customers can build their own virtual networks using compute, memory, and networking resources provided by CSPI. One or more customer resources or workloads, such as compute instances, can be deployed on these virtual networks. For example, a customer can use resources provided by CSPI to build one or multiple customizable and private virtual network(s) referred to as virtual cloud networks (VCNs). A customer can deploy one or more customer resources, such as compute instances, on a customer VCN. Compute instances can take the form of virtual machines, bare metal instances, and the like. The CSPI thus provides infrastructure and a set of complementary cloud services that enable customers to build and run a wide range of applications and services in a highly available virtual hosted environment. The customer does not manage or control the underlying physical resources provided by CSPI but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., firewalls).

**[0045]** The CSP may provide a console that enables customers and network administrators to configure, access, and manage resources deployed in the cloud using CSPI resources. In certain embodiments, the console provides a web-based user interface that can be used to access and manage CSPI. In some implementations, the console is a web-based application provided by the CSP.

**[0046]** CSPI may support single-tenancy or multi-tenancy architectures. In a single tenancy architecture, a software (e.g., an application, a database) or a hardware component (e.g., a host machine or a server) serves a single customer or tenant. In a multi-tenancy architecture, a software or a hardware component serves multiple customers or tenants. Thus, in a multi-tenancy architecture, CSPI resources are shared between multiple customers or tenants. In a multi-tenancy situation, precautions are taken, and safeguards put in place within CSPI to ensure that each tenant's data is isolated and remains invisible to other tenants.

**[0047]** In a physical network, a network endpoint ("endpoint") refers to a computing device or system that is connected to a physical network and communicates back and forth with the network to which it is connected. A network endpoint in the physical network may be

connected to a Local Area Network (LAN), a Wide Area Network (WAN), or other type of physical network. Examples of traditional endpoints in a physical network include modems, hubs, bridges, switches, routers, and other networking devices, physical computers (or host machines), and the like. Each physical device in the physical network has a fixed network address that can be used to communicate with the device. This fixed network address can be a Layer-2 address (e.g., a MAC address), a fixed Layer-3 address (e.g., an IP address), and the like. In a virtualized environment or in a virtual network, the endpoints can include various virtual endpoints such as virtual machines that are hosted by components of the physical network (e.g., hosted by physical host machines). These endpoints in the virtual network are addressed by overlay addresses such as overlay Layer-2 addresses (e.g., overlay MAC addresses) and overlay Layer-3 addresses (e.g., overlay IP addresses). Network overlays enable flexibility by allowing network managers to move around the overlay addresses associated with network endpoints using software management (e.g., via software implementing a control plane for the virtual network). Accordingly, unlike in a physical network, in a virtual network, an overlay address (e.g., an overlay IP address) can be moved from one endpoint to another using network management software. Since the virtual network is built on top of a physical network, communications between components in the virtual network involves both the virtual network and the underlying physical network. In order to facilitate such communications, the components of CSPI are configured to learn and store mappings that map overlay addresses in the virtual network to actual physical addresses in the substrate network, and vice versa. These mappings are then used to facilitate the communications. Customer traffic is encapsulated to facilitate routing in the virtual network.

**[0048]** Accordingly, physical addresses (e.g., physical IP addresses) are associated with components in physical networks and overlay addresses (e.g., overlay IP addresses) are associated with entities in virtual or overlay networks. A physical IP address is an IP address associated with a physical device (e.g., a network device) in the substrate or physical network. For example, each NVD has an associated physical IP address. An overlay IP address is an overlay address associated with an entity in an overlay network, such as with a compute instance in a customer's virtual cloud network (VCN). Two different customers or tenants, each with their own private VCNs can potentially use the same overlay IP address in their VCNs without any knowledge of each other. Both the physical IP addresses and overlay IP addresses are types of real IP addresses. These are separate from virtual IP addresses. A virtual IP address is typically a single IP address that is represents or maps to multiple real IP

addresses. A virtual IP address provides a 1-to-many mapping between the virtual IP address and multiple real IP addresses. For example, a load balancer may use a VIP to map to or represent multiple servers, each server having its own real IP address.

**[0049]** The cloud infrastructure or CSPI is physically hosted in one or more data centers in one or more regions around the world. The CSPI may include components in the physical or substrate network and virtualized components (e.g., virtual networks, compute instances, virtual machines, etc.) that are in a virtual network built on top of the physical network components. In certain embodiments, the CSPI is organized and hosted in realms, regions, and availability domains. A region is typically a localized geographic area that contains one or more data centers. Regions are generally independent of each other and can be separated by vast distances, for example, across countries or even continents. For example, a first region may be in Australia, another one in Japan, yet another one in India, and the like. CSPI resources are divided among regions such that each region has its own independent subset of CSPI resources. Each region may provide a set of core infrastructure services and resources, such as, compute resources (e.g., bare metal servers, virtual machine, containers and related infrastructure, etc.); storage resources (e.g., block volume storage, file storage, object storage, archive storage); networking resources (e.g., virtual cloud networks (VCNs), load balancing resources, connections to on-premise networks), database resources; edge networking resources (e.g., DNS); and access management and monitoring resources, and others. Each region generally has multiple paths connecting it to other regions in the realm.

**[0050]** Generally, an application is deployed in a region (i.e., deployed on infrastructure associated with that region) where it is most heavily used, because using nearby resources is faster than using distant resources. Applications can also be deployed in different regions for various reasons, such as redundancy to mitigate the risk of region-wide events such as large weather systems or earthquakes, to meet varying requirements for legal jurisdictions, tax domains, and other business or social criteria, and the like.

**[0051]** The data centers within a region can be further organized and subdivided into availability domains (ADs). An availability domain may correspond to one or more data centers located within a region. A region can be composed of one or more availability domains. In such a distributed environment, CSPI resources are either region-specific, such as a virtual cloud network (VCN), or availability domain-specific, such as a compute instance.

**[0052]** ADs within a region are isolated from each other, fault tolerant, and are configured such that they are very unlikely to fail simultaneously. This is achieved by the ADs not sharing

critical infrastructure resources such as networking, physical cables, cable paths, cable entry points, etc., such that a failure at one AD within a region is unlikely to impact the availability of the other ADs within the same region. The ADs within the same region may be connected to each other by a low latency, high bandwidth network, which makes it possible to provide high-availability connectivity to other networks (e.g., the Internet, customers' on-premises networks, etc.) and to build replicated systems in multiple ADs for both high-availability and disaster recovery. Cloud services use multiple ADs to ensure high availability and to protect against resource failure. As the infrastructure provided by the IaaS provider grows, more regions and ADs may be added with additional capacity. Traffic between availability domains is usually encrypted.

**[0053]** In certain embodiments, regions are grouped into realms. A realm is a logical collection of regions. Realms are isolated from each other and do not share any data. Regions in the same realm may communicate with each other, but regions in different realms cannot. A customer's tenancy or account with the CSP exists in a single realm and can be spread across one or more regions that belong to that realm. Typically, when a customer subscribes to an IaaS service, a tenancy or account is created for that customer in the customer-specified region (referred to as the "home" region) within a realm. A customer can extend the customer's tenancy across one or more other regions within the realm. A customer cannot access regions that are not in the realm where the customer's tenancy exists.

**[0054]** An IaaS provider can provide multiple realms, each realm catered to a particular set of customers or users. For example, a commercial realm may be provided for commercial customers. As another example, a realm may be provided for a specific country for customers within that country. As yet another example, a government realm may be provided for a government, and the like. For example, the government realm may be catered for a specific government and may have a heightened level of security than a commercial realm. For example, Oracle Cloud Infrastructure (OCI) currently offers a realm for commercial regions and two realms (e.g., FedRAMP authorized and IL5 authorized) for government cloud regions.

**[0055]** In certain embodiments, an AD can be subdivided into one or more fault domains. A fault domain is a grouping of infrastructure resources within an AD to provide anti-affinity. Fault domains allow for the distribution of compute instances such that the instances are not on the same physical hardware within a single AD. This is known as anti-affinity. A fault domain refers to a set of hardware components (computers, switches, and more) that share a single point of failure. A compute pool is logically divided up into fault domains. Due to this,

a hardware failure or compute hardware maintenance event that affects one fault domain does not affect instances in other fault domains. Depending on the embodiment, the number of fault domains for each AD may vary. For instance, in certain embodiments each AD contains three fault domains. A fault domain acts as a logical data center within an AD.

**[0056]** When a customer subscribes to an IaaS service, resources from CSPI are provisioned for the customer and associated with the customer's tenancy. The customer can use these provisioned resources to build private networks and deploy resources on these networks. The customer networks that are hosted in the cloud by the CSPI are referred to as virtual cloud networks (VCNs). A customer can set up one or more virtual cloud networks (VCNs) using CSPI resources allocated for the customer. A VCN is a virtual or software defined private network. The customer resources that are deployed in the customer's VCN can include compute instances (e.g., virtual machines, bare-metal instances) and other resources. These compute instances may represent various customer workloads such as applications, load balancers, databases, and the like. A compute instance deployed on a VCN can communicate with publicly accessible endpoints ("public endpoints") over a public network such as the Internet, with other instances in the same VCN or other VCNs (e.g., the customer's other VCNs, or VCNs not belonging to the customer), with the customer's on-premise data centers or networks, and with service endpoints, and other types of endpoints.

**[0057]** The CSP may provide various services using the CSPI. In some instances, customers of CSPI may themselves act like service providers and provide services using CSPI resources. A service provider may expose a service endpoint, which is characterized by identification information (e.g., an IP Address, a DNS name and port). A customer's resource (e.g., a compute instance) can consume a particular service by accessing a service endpoint exposed by the service for that particular service. These service endpoints are generally endpoints that are publicly accessible by users using public IP addresses associated with the endpoints via a public communication network such as the Internet. Network endpoints that are publicly accessible are also sometimes referred to as public endpoints.

**[0058]** In certain embodiments, a service provider may expose a service via an endpoint (sometimes referred to as a service endpoint) for the service. Customers of the service can then use this service endpoint to access the service. In certain implementations, a service endpoint provided for a service can be accessed by multiple customers that intend to consume that service. In other implementations, a dedicated service endpoint may be provided for a customer such that only that customer can access the service using that dedicated service endpoint.

**[0059]** In certain embodiments, when a VCN is created, it is associated with a private overlay Classless Inter-Domain Routing (CIDR) address space, which is a range of private overlay IP addresses that are assigned to the VCN (e.g., 10.0/16). A VCN includes associated subnets, route tables, and gateways. A VCN resides within a single region but can span one or more or all of the region's availability domains. A gateway is a virtual interface that is configured for a VCN and enables communication of traffic to and from the VCN to one or more endpoints outside the VCN. One or more different types of gateways may be configured for a VCN to enable communication to and from different types of endpoints.

**[0060]** A VCN can be subdivided into one or more sub-networks such as one or more subnets. A subnet is thus a unit of configuration or a subdivision that can be created within a VCN. A VCN can have one or multiple subnets. Each subnet within a VCN is associated with a contiguous range of overlay IP addresses (e.g., 10.0.0.0/24 and 10.0.1.0/24) that do not overlap with other subnets in that VCN, and which represent an address space subset within the address space of the VCN.

**[0061]** Each compute instance is associated with a virtual network interface card (VNIC), that enables the compute instance to participate in a subnet of a VCN. A VNIC is a logical representation of physical Network Interface Card (NIC). In general. a VNIC is an interface between an entity (e.g., a compute instance, a service) and a virtual network. A VNIC exists in a subnet, has one or more associated IP addresses, and associated security rules or policies. A VNIC is equivalent to a Layer-2 port on a switch. A VNIC is attached to a compute instance and to a subnet within a VCN. A VNIC associated with a compute instance enables the compute instance to be a part of a subnet of a VCN and enables the compute instance to communicate (e.g., send and receive packets) with endpoints that are on the same subnet as the compute instance, with endpoints in different subnets in the VCN, or with endpoints outside the VCN. The VNIC associated with a compute instance thus determines how the compute instance connects with endpoints inside and outside the VCN. A VNIC for a compute instance is created and associated with that compute instance when the compute instance is created and added to a subnet within a VCN. For a subnet comprising a set of compute instances, the subnet contains the VNICs corresponding to the set of compute instances, each VNIC attached to a compute instance within the set of computer instances.

**[0062]** Each compute instance is assigned a private overlay IP address via the VNIC associated with the compute instance. This private overlay IP address is assigned to the VNIC that is associated with the compute instance when the compute instance is created and used for

routing traffic to and from the compute instance. All VNICs in a given subnet use the same route table, security lists, and DHCP options. As described above, each subnet within a VCN is associated with a contiguous range of overlay IP addresses (e.g., 10.0.0.0/24 and 10.0.1.0/24) that do not overlap with other subnets in that VCN, and which represent an address space subset within the address space of the VCN. For a VNIC on a particular subnet of a VCN, the private overlay IP address that is assigned to the VNIC is an address from the contiguous range of overlay IP addresses allocated for the subnet.

**[0063]** In certain embodiments, a compute instance may optionally be assigned additional overlay IP addresses in addition to the private overlay IP address, such as, for example, one or more public IP addresses if in a public subnet. These multiple addresses are assigned either on the same VNIC or over multiple VNICs that are associated with the compute instance. Each instance however has a primary VNIC that is created during instance launch and is associated with the overlay private IP address assigned to the instance—this primary VNIC cannot be removed. Additional VNICs, referred to as secondary VNICs, can be added to an existing instance in the same availability domain as the primary VNIC. All the VNICs are in the same availability domain as the instance. A secondary VNIC can be in a subnet in the same VCN as the primary VNIC, or in a different subnet that is either in the same VCN or a different one.

**[0064]** A compute instance may optionally be assigned a public IP address if it is in a public subnet. A subnet can be designated as either a public subnet or a private subnet at the time the subnet is created. A private subnet means that the resources (e.g., compute instances) and associated VNICs in the subnet cannot have public overlay IP addresses. A public subnet means that the resources and associated VNICs in the subnet can have public IP addresses. A customer can designate a subnet to exist either in a single availability domain or across multiple availability domains in a region or realm.

**[0065]** As described above, a VCN may be subdivided into one or more subnets. In certain embodiments, a Virtual Router (VR) configured for the VCN (referred to as the VCN VR or just VR) enables communications between the subnets of the VCN. For a subnet within a VCN, the VR represents a logical gateway for that subnet that enables the subnet (i.e., the compute instances on that subnet) to communicate with endpoints on other subnets within the VCN, and with other endpoints outside the VCN. The VCN VR is a logical entity that is configured to route traffic between VNICs in the VCN and virtual gateways ("gateways") associated with the VCN. Gateways are further described below with respect to FIG. 1. A VCN VR is a Layer-3/IP Layer concept. In one embodiment, there is one VCN VR for a VCN

where the VCN VR has potentially an unlimited number of ports addressed by IP addresses, with one port for each subnet of the VCN. In this manner, the VCN VR has a different IP address for each subnet in the VCN that the VCN VR is attached to. The VR is also connected to the various gateways configured for a VCN. In certain embodiments, a particular overlay IP address from the overlay IP address range for a subnet is reserved for a port of the VCN VR for that subnet. For example, consider a VCN having two subnets with associated address ranges 10.0/16 and 10.1/16, respectively. For the first subnet within the VCN with address range 10.0/16, an address from this range is reserved for a port of the VCN VR for that subnet. In some instances, the first IP address from the range may be reserved for the VCN VR. For example, for the subnet with overlay IP address range 10.0/16, IP address 10.0.0.1 may be reserved for a port of the VCN VR for that subnet. For the second subnet within the same VCN with address range 10.1/16, the VCN VR may have a port for that second subnet with IP address 10.1.0.1. The VCN VR has a different IP address for each of the subnets in the VCN.

**[0066]** In some other embodiments, each subnet within a VCN may have its own associated VR that is addressable by the subnet using a reserved or default IP address associated with the VR. The reserved or default IP address may, for example, be the first IP address from the range of IP addresses associated with that subnet. The VNICs in the subnet can communicate (e.g., send and receive packets) with the VR associated with the subnet using this default or reserved IP address. In such an embodiment, the VR is the ingress/egress point for that subnet. The VR associated with a subnet within the VCN can communicate with other VRs associated with other subnets within the VCN. The VRs can also communicate with gateways associated with the VCN. The VR function for a subnet is running on or executed by one or more NVDs executing VNICs functionality for VNICs in the subnet.

**[0067]** Route tables, security rules, and DHCP options may be configured for a VCN. Route tables are virtual route tables for the VCN and include rules to route traffic from subnets within the VCN to destinations outside the VCN by way of gateways or specially configured instances. A VCN's route tables can be customized to control how packets are forwarded/routed to and from the VCN. DHCP options refers to configuration information that is automatically provided to the instances when they boot up.

**[0068]** Security rules configured for a VCN represent overlay firewall rules for the VCN. The security rules can include ingress and egress rules, and specify the types of traffic (e.g., based upon protocol and port) that is allowed in and out of the instances within the VCN. The customer can choose whether a given rule is stateful or stateless. For instance, the customer

can allow incoming SSH traffic from anywhere to a set of instances by setting up a stateful ingress rule with source CIDR 0.0.0.0/0, and destination TCP port 22. Security rules can be implemented using network security groups or security lists. A network security group consists of a set of security rules that apply only to the resources in that group. A security list, on the other hand, includes rules that apply to all the resources in any subnet that uses the security list. A VCN may be provided with a default security list with default security rules. DHCP options configured for a VCN provide configuration information that is automatically provided to the instances in the VCN when the instances boot up.

**[0069]** In certain embodiments, the configuration information for a VCN is determined and stored by a VCN Control Plane. The configuration information for a VCN may include, for example, information about the address range associated with the VCN, subnets within the VCN and associated information, one or more VRs associated with the VCN, compute instances in the VCN and associated VNICs, NVDs executing the various virtualization network functions (e.g., VNICs, VRs, gateways) associated with the VCN, state information for the VCN, and other VCN-related information. In certain embodiments, a VCN Distribution Service publishes the configuration information stored by the VCN Control Plane, or portions thereof, to the NVDs. The distributed information may be used to update information (e.g., forwarding tables, routing tables, etc.) stored and used by the NVDs to forward packets to and from the compute instances in the VCN.

**[0070]** In certain embodiments, the creation of VCNs and subnets are handled by a VCN Control Plane (CP), and the launching of compute instances is handled by a Compute Control Plane. The Compute Control Plane is responsible for allocating the physical resources for the compute instance and then calls the VCN Control Plane to create and attach VNICs to the compute instance. The VCN CP also sends VCN data mappings to the VCN data plane that is configured to perform packet forwarding and routing functions. In certain embodiments, the VCN CP provides a distribution service that is responsible for providing updates to the VCN data plane. Examples of a VCN Control Plane are also depicted in FIGs. 6, 7, 8, and 9 (see references 616, 716, 816, and 916) and described below.

**[0071]** A customer may create one or more VCNs using resources hosted by CSPI. A compute instance deployed on a customer VCN may communicate with different endpoints. These endpoints can include endpoints that are hosted by CSPI and endpoints outside CSPI.

**[0072]** Various different architectures for implementing cloud-based service using CSPI are depicted in FIGs. 1, 2, 3, 4, 5, and 18-22, and are described below. FIG. 1 is a high-level

diagram of a distributed environment 100 showing an overlay or customer VCN hosted by CSPI according to certain embodiments. The distributed environment depicted in FIG. 1 includes multiple components in the overlay network. Distributed environment 100 depicted in FIG. 1 is merely an example and is not intended to unduly limit the scope of claimed embodiments. Many variations, alternatives, and modifications are possible. For example, in some implementations, the distributed environment depicted in FIG. 1 may have more or fewer systems or components than those shown in FIG. 1, may combine two or more systems, or may have a different configuration or arrangement of systems.

**[0073]** As shown in the example depicted in FIG. 1, distributed environment 100 comprises CSPI 101 that provides services and resources that customers can subscribe to and use to build their virtual cloud networks (VCNs). In certain embodiments, CSPI 101 offers IaaS services to subscribing customers. The data centers within CSPI 101 may be organized into one or more regions. One example region "Region US" 102 is shown in FIG. 1. A customer has configured a customer VCN c/o Oracle International Corporation for region 102. The customer may deploy various compute instances on VCN 104, where the compute instances may include virtual machines or bare metal instances. Examples of instances include applications, database, load balancers, and the like.

**[0074]** In the embodiment depicted in FIG. 1, customer VCN 104 comprises two subnets, namely, "Subnet-1" and "Subnet-2", each subnet with its own CIDR IP address range. In FIG. 1, the overlay IP address range for Subnet-1 is 10.0/16 and the address range for Subnet-2 is 10.1/16. A VCN Virtual Router 105 represents a logical gateway for the VCN that enables communications between subnets of the VCN 104, and with other endpoints outside the VCN. VCN VR 105 is configured to route traffic between VNICs in VCN 104 and gateways associated with VCN 104. VCN VR 105 provides a port for each subnet of VCN 104. For example, VR 105 may provide a port with IP address 10.0.0.1 for Subnet-1 and a port with IP address 10.1.0.1 for Subnet-2.

**[0075]** Multiple compute instances may be deployed on each subnet, where the compute instances can be virtual machine instances, and/or bare metal instances. The compute instances in a subnet may be hosted by one or more host machines within CSPI 101. A compute instance participates in a subnet via a VNIC associated with the compute instance. For example, as shown in FIG. 1, a compute instance C1 is part of Subnet-1 via a VNIC associated with the compute instance. Likewise, compute instance C2 is part of Subnet-1 via a VNIC associated with C2. In a similar manner, multiple compute instances, which may be virtual machine

instances or bare metal instances, may be part of Subnet-1. Via its associated VNIC, each compute instance is assigned a private overlay IP address and a MAC address. For example, in FIG. 1, compute instance C1 has an overlay IP address of 10.0.0.2 and a MAC address of M1, while compute instance C2 has a private overlay IP address of 10.0.0.3 and a MAC address of M2. Each compute instance in Subnet-1, including compute instances C1 and C2, has a default route to VCN VR 105 using IP address 10.0.0.1, which is the IP address for a port of VCN VR 105 for Subnet-1.

**[0076]** Subnet-2 can have multiple compute instances deployed on it, including virtual machine instances and/or bare metal instances. For example, as shown in FIG. 1, compute instances D1 and D2 are part of Subnet-2 via VNICs associated with the respective compute instances. In the embodiment depicted in FIG. 1, compute instance D1 has an overlay IP address of 10.1.0.2 and a MAC address of MM1, while compute instance D2 has a private overlay IP address of 10.1.0.3 and a MAC address of MM2. Each compute instance in Subnet-2, including compute instances D1 and D2, has a default route to VCN VR 105 using IP address 10.1.0.1, which is the IP address for a port of VCN VR 105 for Subnet-2.

**[0077]** VCN A 104 may also include one or more load balancers. For example, a load balancer may be provided for a subnet and may be configured to load balance traffic across multiple compute instances on the subnet. A load balancer may also be provided to load balance traffic across subnets in the VCN.

**[0078]** A particular compute instance deployed on VCN 104 can communicate with various different endpoints. These endpoints may include endpoints that are hosted by CSPI 200 and endpoints outside CSPI 200. Endpoints that are hosted by CSPI 101 may include: an endpoint on the same subnet as the particular compute instance (e.g., communications between two compute instances in Subnet-1); an endpoint on a different subnet but within the same VCN (e.g., communication between a compute instance in Subnet-1 and a compute instance in Subnet-2); an endpoint in a different VCN in the same region (e.g., communications between a compute instance in Subnet-1 and an endpoint in a VCN in the same region 106 or 110, communications between a compute instance in Subnet-1 and an endpoint in service network 110 in the same region); or an endpoint in a VCN in a different region (e.g., communications between a compute instance in Subnet-1 and an endpoint in a VCN in a different region 108). A compute instance in a subnet hosted by CSPI 101 may also communicate with endpoints that are not hosted by CSPI 101 (i.e., are outside CSPI 101). These outside endpoints include endpoints in the customer's on-premises network 116, endpoints within other remote cloud

hosted networks 118, public endpoints 114 accessible via a public network such as the Internet, and other endpoints.

**[0079]** Communications between compute instances on the same subnet are facilitated using VNICs associated with the source compute instance and the destination compute instance. For example, compute instance C1 in Subnet-1 may want to send packets to compute instance C2 in Subnet-1. For a packet originating at a source compute instance and whose destination is another compute instance in the same subnet, the packet is first processed by the VNIC associated with the source compute instance. Processing performed by the VNIC associated with the source compute instance can include determining destination information for the packet from the packet headers, identifying any policies (e.g., security lists) configured for the VNIC associated with the source compute instance, determining a next hop for the packet, performing any packet encapsulation/decapsulation functions as needed, and then forwarding/routing the packet to the next hop with the goal of facilitating communication of the packet to its intended destination. When the destination compute instance is in the same subnet as the source compute instance, the VNIC associated with the source compute instance is configured to identify the VNIC associated with the destination compute instance and forward the packet to that VNIC for processing. The VNIC associated with the destination compute instance is then executed and forwards the packet to the destination compute instance.

**[0080]** For a packet to be communicated from a compute instance in a subnet to an endpoint in a different subnet in the same VCN, the communication is facilitated by the VNICs associated with the source and destination compute instances and the VCN VR. For example, if compute instance C1 in Subnet-1 in FIG. 1 wants to send a packet to compute instance D1 in Subnet-2, the packet is first processed by the VNIC associated with compute instance C1. The VNIC associated with compute instance C1 is configured to route the packet to the VCN VR 105 using default route or port 10.0.0.1 of the VCN VR. VCN VR 105 is configured to route the packet to Subnet-2 using port 10.1.0.1. The packet is then received and processed by the VNIC associated with D1 and the VNIC forwards the packet to compute instance D1.

**[0081]** For a packet to be communicated from a compute instance in VCN 104 to an endpoint that is outside VCN 104, the communication is facilitated by the VNIC associated with the source compute instance, VCN VR 105, and gateways associated with VCN 104. One or more types of gateways may be associated with VCN 104. A gateway is an interface between a VCN and another endpoint, where another endpoint is outside the VCN. A gateway is a Layer-3/IP layer concept and enables a VCN to communicate with endpoints outside the VCN. A gateway

thus facilitates traffic flow between a VCN and other VCNs or networks. Various different types of gateways may be configured for a VCN to facilitate different types of communications with different types of endpoints. Depending upon the gateway, the communications may be over public networks (e.g., the Internet) or over private networks. Various communication protocols may be used for these communications.

**[0082]** For example, compute instance C1 may want to communicate with an endpoint outside VCN 104. The packet may be first processed by the VNIC associated with source compute instance C1. The VNIC processing determines that the destination for the packet is outside the Subnet-1 of C1. The VNIC associated with C1 may forward the packet to VCN VR 105 for VCN 104. VCN VR 105 then processes the packet and as part of the processing, based upon the destination for the packet, determines a particular gateway associated with VCN 104 as the next hop for the packet. VCN VR 105 may then forward the packet to the particular identified gateway. For example, if the destination is an endpoint within the customer's on-premise network, then the packet may be forwarded by VCN VR 105 to Dynamic Routing Gateway (DRG) gateway 122 configured for VCN 104. The packet may then be forwarded from the gateway to a next hop to facilitate communication of the packet to it final intended destination.

**[0083]** Various different types of gateways may be configured for a VCN. Examples of gateways that may be configured for a VCN are depicted in FIG. 1 and described below. Examples of gateways associated with a VCN are also depicted in FIGs. 18, 19, 20, and 21 (for example, gateways referenced by reference numbers 1834, 1836, 1838, 1934, 1936, 1938, 2034, 2036, 2038, 2134, 2136, and 2138) and described below. As shown in the embodiment depicted in FIG. 1, a Dynamic Routing Gateway (DRG) 122 may be added to or be associated with customer VCN 104 and provides a path for private network traffic communication between customer VCN 104 and another endpoint, where the another endpoint can be the customer's on-premise network 116, a VCN 108 in a different region of CSPI 101, or other remote cloud networks 118 not hosted by CSPI 101. Customer on-premise network 116 may be a customer network or a customer data center built using the customer's resources. Access to customer on-premise network 116 is generally very restricted. For a customer that has both a customer on-premise network 116 and one or more VCNs 104 deployed or hosted in the cloud by CSPI 101, the customer may want their on-premise network 116 and their cloud based VCN 104 to be able to communicate with each other. This enables a customer to build an extended hybrid environment encompassing the customer's VCN 104 hosted by CSPI 101 and

their on-premises network 116. DRG 122 enables this communication. To enable such communications, a communication channel 124 is set up where one endpoint of the channel is in customer on-premise network 116 and the other endpoint is in CSPI 101 and connected to customer VCN 104. Communication channel 124 can be over public communication networks such as the Internet or private communication networks. Various different communication protocols may be used such as IPsec VPN technology over a public communication network such as the Internet, Oracle's FastConnect technology that uses a private network instead of a public network, and others. The device or equipment in customer on-premise network 116 that forms one end point for communication channel 124 is referred to as the customer premise equipment (CPE), such as CPE 126 depicted in FIG. 1. On the CSPI 101 side, the endpoint may be a host machine executing DRG 122.

**[0084]** In certain embodiments, a Remote Peering Connection (RPC) can be added to a DRG, which allows a customer to peer one VCN with another VCN in a different region. Using such an RPC, customer VCN 104 can use DRG 122 to connect with a VCN 108 in another region. DRG 122 may also be used to communicate with other remote cloud networks 118, not hosted by CSPI 101 such as a Microsoft Azure cloud, Amazon AWS cloud, and others.

**[0085]** As shown in FIG. 1, an Internet Gateway (IGW) 120 may be configured for customer VCN 104 the enables a compute instance on VCN 104 to communicate with public endpoints 114 accessible over a public network such as the Internet. IGW 120 is a gateway that connects a VCN to a public network such as the Internet. IGW 120 enables a public subnet (where the resources in the public subnet have public overlay IP addresses) within a VCN, such as VCN 104, direct access to public endpoints 112 on a public network 114 such as the Internet. Using IGW 120, connections can be initiated from a subnet within VCN 104 or from the Internet.

**[0086]** A Network Address Translation (NAT) gateway 128 can be configured for customer's VCN 104 and enables cloud resources in the customer's VCN, which do not have dedicated public overlay IP addresses, access to the Internet and it does so without exposing those resources to direct incoming Internet connections (e.g., L4-L7 connections). This enables a private subnet within a VCN, such as private Subnet-1 in VCN 104, with private access to public endpoints on the Internet. In NAT gateways, connections can be initiated only from the private subnet to the public Internet and not from the Internet to the private subnet.

**[0087]** In certain embodiments, a Service Gateway (SGW) 126 can be configured for customer VCN 104 and provides a path for private network traffic between VCN 104 and supported services endpoints in a service network 110. In certain embodiments, service

network 110 may be provided by the CSP and may provide various services. An example of such a service network is Oracle's Services Network, which provides various services that can be used by customers. For example, a compute instance (e.g., a database system) in a private subnet of customer VCN 104 can back up data to a service endpoint (e.g., Object Storage) without needing public IP addresses or access to the Internet. In certain embodiments, a VCN can have only one SGW, and connections can only be initiated from a subnet within the VCN and not from service network 110. If a VCN is peered with another, resources in the other VCN typically cannot access the SGW. Resources in on-premises networks that are connected to a VCN with FastConnect or VPN Connect can also use the service gateway configured for that VCN.

**[0088]** In certain implementations, SGW 126 uses the concept of a service Classless Inter-Domain Routing (CIDR) label, which is a string that represents all the regional public IP address ranges for the service or group of services of interest. The customer uses the service CIDR label when they configure the SGW and related route rules to control traffic to the service. The customer can optionally utilize it when configuring security rules without needing to adjust them if the service's public IP addresses change in the future.

**[0089]** A Local Peering Gateway (LPG) 132 is a gateway that can be added to customer VCN 104 and enables VCN 104 to peer with another VCN in the same region. Peering means that the VCNs communicate using private IP addresses, without the traffic traversing a public network such as the Internet or without routing the traffic through the customer's on-premises network 116. In preferred embodiments, a VCN has a separate LPG for each peering it establishes. Local Peering or VCN Peering is a common practice used to establish network connectivity between different applications or infrastructure management functions.

**[0090]** Service providers, such as providers of services in service network 110, may provide access to services using different access models. According to a public access model, services may be exposed as public endpoints that are publicly accessible by compute instance in a customer VCN via a public network such as the Internet and or may be privately accessible via SGW 126. According to a specific private access model, services are made accessible as private IP endpoints in a private subnet in the customer's VCN. This is referred to as a Private Endpoint (PE) access and enables a service provider to expose their service as an instance in the customer's private network. A Private Endpoint resource represents a service within the customer's VCN. Each PE manifests as a VNIC (referred to as a PE-VNIC, with one or more private IPs) in a subnet chosen by the customer in the customer's VCN. A PE thus provides a

way to present a service within a private customer VCN subnet using a VNIC. Since the endpoint is exposed as a VNIC, all the features associates with a VNIC such as routing rules, security lists, etc., are now available for the PE VNIC.

**[0091]** A service provider can register their service to enable access through a PE. The provider can associate policies with the service that restricts the service's visibility to the customer tenancies. A provider can register multiple services under a single virtual IP address (VIP), especially for multi-tenant services. There may be multiple such private endpoints (in multiple VCNs) that represent the same service.

**[0092]** Compute instances in the private subnet can then use the PE VNIC's private IP address or the service DNS name to access the service. Compute instances in the customer VCN can access the service by sending traffic to the private IP address of the PE in the customer VCN. A Private Access Gateway (PAGW) 130 is a gateway resource that can be attached to a service provider VCN (e.g., a VCN in service network 110) that acts as an ingress/egress point for all traffic from/to customer subnet private endpoints. PAGW 130 enables a provider to scale the number of PE connections without utilizing its internal IP address resources. A provider needs only configure one PAGW for any number of services registered in a single VCN. Providers can represent a service as a private endpoint in multiple VCNs of one or more customers. From the customer's perspective, the PE VNIC, which, instead of being attached to a customer's instance, appears attached to the service with which the customer wishes to interact. The traffic destined to the private endpoint is routed via PAGW 130 to the service. These are referred to as customer-to-service private connections (C2S connections).

**[0093]** The PE concept can also be used to extend the private access for the service to customer's on-premises networks and data centers, by allowing the traffic to flow through FastConnect/IPsec links and the private endpoint in the customer VCN. Private access for the service can also be extended to the customer's peered VCNs, by allowing the traffic to flow between LPG 132 and the PE in the customer's VCN.

**[0094]** A customer can control routing in a VCN at the subnet level, so the customer can specify which subnets in the customer's VCN, such as VCN 104, use each gateway. A VCN's route tables are used to decide if traffic is allowed out of a VCN through a particular gateway. For example, in a particular instance, a route table for a public subnet within customer VCN 104 may send non-local traffic through IGW 120. The route table for a private subnet within the same customer VCN 104 may send traffic destined for CSP services through SGW 126.

All remaining traffic may be sent via the NAT gateway 128. Route tables only control traffic going out of a VCN.

**[0095]** Security lists associated with a VCN are used to control traffic that comes into a VCN via a gateway via inbound connections. All resources in a subnet use the same route table and security lists. Security lists may be used to control specific types of traffic allowed in and out of instances in a subnet of a VCN. Security list rules may comprise ingress (inbound) and egress (outbound) rules. For example, an ingress rule may specify an allowed source address range, while an egress rule may specify an allowed destination address range. Security rules may specify a particular protocol (e.g., TCP, ICMP), a particular port (e.g., 22 for SSH, 3389 for Windows RDP), etc. In certain implementations, an instance's operating system may enforce its own firewall rules that are aligned with the security list rules. Rules may be stateful (e.g., a connection is tracked, and the response is automatically allowed without an explicit security list rule for the response traffic) or stateless.

**[0096]** Access from a customer VCN (i.e., by a resource or compute instance deployed on VCN 104) can be categorized as public access, private access, or dedicated access. Public access refers to an access model where a public IP address or a NAT is used to access a public endpoint. Private access enables customer workloads in VCN 104 with private IP addresses (e.g., resources in a private subnet) to access services without traversing a public network such as the Internet. In certain embodiments, CSPI 101 enables customer VCN workloads with private IP addresses to access the (public service endpoints of) services using a service gateway. A service gateway thus offers a private access model by establishing a virtual link between the customer's VCN and the service's public endpoint residing outside the customer's private network.

**[0097]** Additionally, CSPI may offer dedicated public access using technologies such as FastConnect public peering where customer on-premises instances can access one or more services in a customer VCN using a FastConnect connection and without traversing a public network such as the Internet. CSPI also may also offer dedicated private access using FastConnect private peering where customer on-premises instances with private IP addresses can access the customer's VCN workloads using a FastConnect connection. FastConnect is a network connectivity alternative to using the public Internet to connect a customer's on-premise network to CSPI and its services. FastConnect provides an easy, elastic, and economical way to create a dedicated and private connection with higher bandwidth options and a more reliable and consistent networking experience when compared to Internet-based connections.

**[0098]** FIG. 1 and the accompanying description above describes various virtualized components in an example virtual network. As described above, the virtual network is built on the underlying physical or substrate network. FIG. 2 depicts a simplified architectural diagram of the physical components in the physical network within CSPI 200 that provide the underlay for the virtual network according to certain embodiments. As shown, CSPI 200 provides a distributed environment comprising components and resources (e.g., compute, memory, and networking resources) provided by a cloud service provider (CSP). These components and resources are used to provide cloud services (e.g., IaaS services) to subscribing customers, i.e., customers that have subscribed to one or more services provided by the CSP. Based upon the services subscribed to by a customer, a subset of resources (e.g., compute, memory, and networking resources) of CSPI 200 are provisioned for the customer. Customers can then build their own cloud-based (i.e., CSPI-hosted) customizable and private virtual networks using physical compute, memory, and networking resources provided by CSPI 200. As previously indicated, these customer networks are referred to as virtual cloud networks (VCNs). A customer can deploy one or more customer resources, such as compute instances, on these customer VCNs. Compute instances can be in the form of virtual machines, bare metal instances, and the like. CSPI 200 provides infrastructure and a set of complementary cloud services that enable customers to build and run a wide range of applications and services in a highly available hosted environment.

**[0099]** In the example embodiment depicted in FIG. 2, the physical components of CSPI 200 include one or more physical host machines or physical servers (e.g., 202, 206, 208), network virtualization devices (NVDs) (e.g., 210, 212), top-of-rack (TOR) switches (e.g., 214, 216), and a physical network (e.g., 218), and switches in physical network 218. The physical host machines or servers may host and execute various compute instances that participate in one or more subnets of a VCN. The compute instances may include virtual machine instances, and bare metal instances. For example, the various compute instances depicted in FIG. 1 may be hosted by the physical host machines depicted in FIG. 2. The virtual machine compute instances in a VCN may be executed by one host machine or by multiple different host machines. The physical host machines may also host virtual host machines, container-based hosts or functions, and the like. The VNICs and VCN VR depicted in FIG. 1 may be executed by the NVDs depicted in FIG. 2. The gateways depicted in FIG. 1 may be executed by the host machines and/or by the NVDs depicted in FIG. 2.

**[0100]** The host machines or servers may execute a hypervisor (also referred to as a virtual machine monitor or VMM) that creates and enables a virtualized environment on the host machines. The virtualization or virtualized environment facilitates cloud-based computing. One or more compute instances may be created, executed, and managed on a host machine by a hypervisor on that host machine. The hypervisor on a host machine enables the physical computing resources of the host machine (e.g., compute, memory, and networking resources) to be shared between the various compute instances executed by the host machine.

**[0101]** For example, as depicted in FIG. 2, host machines 202 and 208 execute hypervisors 260 and 266, respectively. These hypervisors may be implemented using software, firmware, or hardware, or combinations thereof. Typically, a hypervisor is a process or a software layer that sits on top of the host machine's operating system (OS), which in turn executes on the hardware processors of the host machine. The hypervisor provides a virtualized environment by enabling the physical computing resources (e.g., processing resources such as processors/cores, memory resources, networking resources) of the host machine to be shared among the various virtual machine compute instances executed by the host machine. For example, in FIG. 2, hypervisor 260 may sit on top of the OS of host machine 202 and enables the computing resources (e.g., processing, memory, and networking resources) of host machine 202 to be shared between compute instances (e.g., virtual machines) executed by host machine 202. A virtual machine can have its own operating system (referred to as a guest operating system), which may be the same as or different from the OS of the host machine. The operating system of a virtual machine executed by a host machine may be the same as or different from the operating system of another virtual machine executed by the same host machine. A hypervisor thus enables multiple operating systems to be executed alongside each other while sharing the same computing resources of the host machine. The host machines depicted in FIG. 2 may have the same or different types of hypervisors.

**[0102]** A compute instance can be a virtual machine instance or a bare metal instance. In FIG. 2, compute instances 268 on host machine 202 and 274 on host machine 208 are examples of virtual machine instances. Host machine 206 is an example of a bare metal instance that is provided to a customer.

**[0103]** In certain instances, an entire host machine may be provisioned to a single customer, and all of the one or more compute instances (either virtual machines or bare metal instance) hosted by that host machine belong to that same customer. In other instances, a host machine may be shared between multiple customers (i.e., multiple tenants). In such a multi-tenancy

scenario, a host machine may host virtual machine compute instances belonging to different customers. These compute instances may be members of different VCNs of different customers. In certain embodiments, a bare metal compute instance is hosted by a bare metal server without a hypervisor. When a bare metal compute instance is provisioned, a single customer or tenant maintains control of the physical CPU, memory, and network interfaces of the host machine hosting the bare metal instance and the host machine is not shared with other customers or tenants.

**[0104]** As previously described, each compute instance that is part of a VCN is associated with a VNIC that enables the compute instance to become a member of a subnet of the VCN. The VNIC associated with a compute instance facilitates the communication of packets or frames to and from the compute instance. A VNIC is associated with a compute instance when the compute instance is created. In certain embodiments, for a compute instance executed by a host machine, the VNIC associated with that compute instance is executed by an NVD connected to the host machine. For example, in FIG. 2, host machine 202 executes a virtual machine compute instance 268 that is associated with VNIC 276, and VNIC 276 is executed by NVD 210 connected to host machine 202. As another example, bare metal instance 272 hosted by host machine 206 is associated with VNIC 280 that is executed by NVD 212 connected to host machine 206. As yet another example, VNIC 284 is associated with compute instance 274 executed by host machine 208, and VNIC 284 is executed by NVD 212 connected to host machine 208.

**[0105]** For compute instances hosted by a host machine, an NVD connected to that host machine also executes VCN VRs corresponding to VCNs of which the compute instances are members. For example, in the embodiment depicted in FIG. 2, NVD 210 executes VCN VR 277 corresponding to the VCN of which compute instance 268 is a member. NVD 212 may also execute one or more VCN VRs 283 corresponding to VCNs corresponding to the compute instances hosted by host machines 206 and 208.

**[0106]** A host machine may include one or more network interface cards (NIC) that enable the host machine to be connected to other devices. A NIC on a host machine may provide one or more ports (or interfaces) that enable the host machine to be communicatively connected to another device. For example, a host machine may be connected to an NVD using one or more ports (or interfaces) provided on the host machine and on the NVD. A host machine may also be connected to other devices such as another host machine.

**[0107]** For example, in FIG. 2, host machine 202 is connected to NVD 210 using link 220 that extends between a port 234 provided by a NIC 232 of host machine 202 and between a port 236 of NVD 210. Host machine 206 is connected to NVD 212 using link 224 that extends between a port 246 provided by a NIC 244 of host machine 206 and between a port 248 of NVD 212. Host machine 208 is connected to NVD 212 using link 226 that extends between a port 252 provided by a NIC 250 of host machine 208 and between a port 254 of NVD 212.

**[0108]** The NVDs are in turn connected via communication links to top-of-the-rack (TOR) switches, which are connected to physical network 218 (also referred to as the switch fabric). In certain embodiments, the links between a host machine and an NVD, and between an NVD and a TOR switch are Ethernet links. For example, in FIG. 2, NVDs 210 and 212 are connected to TOR switches 214 and 216, respectively, using links 228 and 230. In certain embodiments, the links 220, 224, 226, 228, and 230 are Ethernet links. The collection of host machines and NVDs that are connected to a TOR is sometimes referred to as a rack.

**[0109]** Physical network 218 provides a communication fabric that enables TOR switches to communicate with each other. Physical network 218 can be a multi-tiered network. In certain implementations, physical network 218 is a multi-tiered Clos network of switches, with TOR switches 214 and 216 representing the leaf level nodes of the multi-tiered and multi-node physical switching network 218. Different Clos network configurations are possible including but not limited to a 2-tier network, a 3-tier network, a 4-tier network, a 5-tier network, and in general a "n"-tiered network. An example of a Clos network is depicted in FIG. 5 and described below.

**[0110]** Various different connection configurations are possible between host machines and NVDs such as one-to-one configuration, many-to-one configuration, one-to-many configuration, and others. In a one-to-one configuration implementation, each host machine is connected to its own separate NVD. For example, in FIG. 2, host machine 202 is connected to NVD 210 via NIC 232 of host machine 202. In a many-to-one configuration, multiple host machines are connected to one NVD. For example, in FIG. 2, host machines 206 and 208 are connected to the same NVD 212 via NICs 244 and 250, respectively.

**[0111]** In a one-to-many configuration, one host machine is connected to multiple NVDs. FIG. 3 shows an example within CSPI 300 where a host machine is connected to multiple NVDs. As shown in FIG. 3, host machine 302 comprises a network interface card (NIC) 304 that includes multiple ports 306 and 308. Host machine 300 is connected to a first NVD 310 via port 306 and link 320 and connected to a second NVD 312 via port 308 and link 322. Ports

306 and 308 may be Ethernet ports and the links 320 and 322 between host machine 302 and NVDs 310 and 312 may be Ethernet links. NVD 310 is in turn connected to a first TOR switch 314 and NVD 312 is connected to a second TOR switch 316. The links between NVDs 310 and 312, and TOR switches 314 and 316 may be Ethernet links. TOR switches 314 and 316 represent the Tier-0 switching devices in multi-tiered physical network 318.

**[0112]** The arrangement depicted in FIG. 3 provides two separate physical network paths to and from physical switch network 318 to host machine 302: a first path traversing TOR switch 314 to NVD 310 to host machine 302, and a second path traversing TOR switch 316 to NVD 312 to host machine 302. The separate paths provide for enhanced availability (referred to as high availability) of host machine 302. If there are problems in one of the paths (e.g., a link in one of the paths goes down) or devices (e.g., a particular NVD is not functioning), then the other path may be used for communications to/from host machine 302.

**[0113]** In the configuration depicted in FIG. 3, the host machine is connected to two different NVDs using two different ports provided by a NIC of the host machine. In other embodiments, a host machine may include multiple NICs that enable connectivity of the host machine to multiple NVDs.

**[0114]** Referring back to FIG. 2, an NVD is a physical device or component that performs one or more network and/or storage virtualization functions. An NVD may be any device with one or more processing units (e.g., CPUs, Network Processing Units (NPUs), FPGAs, packet processing pipelines, etc.), memory including cache, and ports. The various virtualization functions may be performed by software/firmware executed by the one or more processing units of the NVD.

**[0115]** An NVD may be implemented in various different forms. For example, in certain embodiments, an NVD is implemented as an interface card referred to as a smartNIC or an intelligent NIC with an embedded processor onboard. A smartNIC is a separate device from the NICs on the host machines. In FIG. 2, the NVDs 210 and 212 may be implemented as smartNICs that are connected to host machines 202, and host machines 206 and 208, respectively.

**[0116]** A smartNIC is however just one example of an NVD implementation. Various other implementations are possible. For example, in some other implementations, an NVD or one or more functions performed by the NVD may be incorporated into or performed by one or more host machines, one or more TOR switches, and other components of CSPI 200. For example, an NVD may be embodied in a host machine where the functions performed by an

NVD are performed by the host machine. As another example, an NVD may be part of a TOR switch, or a TOR switch may be configured to perform functions performed by an NVD that enables the TOR switch to perform various complex packet transformations that are used for a public cloud. A TOR that performs the functions of an NVD is sometimes referred to as a smart TOR. In yet other implementations, where virtual machines (VMs) instances, but not bare metal (BM) instances, are offered to customers, functions performed by an NVD may be implemented inside a hypervisor of the host machine. In some other implementations, some of the functions of the NVD may be offloaded to a centralized service running on a fleet of host machines.

**[0117]** In certain embodiments, such as when implemented as a smartNIC as shown in FIG. 2, an NVD may comprise multiple physical ports that enable it to be connected to one or more host machines and to one or more TOR switches. A port on an NVD can be classified as a host-facing port (also referred to as a "south port") or a network-facing or TOR-facing port (also referred to as a "north port"). A host-facing port of an NVD is a port that is used to connect the NVD to a host machine. Examples of host-facing ports in FIG. 2 include port 236 on NVD 210, and ports 248 and 254 on NVD 212. A network-facing port of an NVD is a port that is used to connect the NVD to a TOR switch. Examples of network-facing ports in FIG. 2 include port 256 on NVD 210, and port 258 on NVD 212. As shown in FIG. 2, NVD 210 is connected to TOR switch 214 using link 228 that extends from port 256 of NVD 210 to the TOR switch 214. Likewise, NVD 212 is connected to TOR switch 216 using link 230 that extends from port 258 of NVD 212 to the TOR switch 216.

**[0118]** An NVD receives packets and frames from a host machine (e.g., packets and frames generated by a compute instance hosted by the host machine) via a host-facing port and, after performing the necessary packet processing, may forward the packets and frames to a TOR switch via a network-facing port of the NVD. An NVD may receive packets and frames from a TOR switch via a network-facing port of the NVD and, after performing the necessary packet processing, may forward the packets and frames to a host machine via a host-facing port of the NVD.

**[0119]** In certain embodiments, there may be multiple ports and associated links between an NVD and a TOR switch. These ports and links may be aggregated to form a link aggregator group of multiple ports or links (referred to as a LAG). Link aggregation allows multiple physical links between two endpoints (e.g., between an NVD and a TOR switch) to be treated as a single logical link. All the physical links in a given LAG may operate in full-duplex mode

at the same speed. LAGs help increase the bandwidth and reliability of the connection between two endpoints. If one of the physical links in the LAG goes down, traffic is dynamically and transparently reassigned to one of the other physical links in the LAG. The aggregated physical links deliver higher bandwidth than each individual link. The multiple ports associated with a LAG are treated as a single logical port. Traffic can be load-balanced across the multiple physical links of a LAG. One or more LAGs may be configured between two endpoints. The two endpoints may be between an NVD and a TOR switch, between a host machine and an NVD, and the like.

**[0120]** An NVD implements or performs network virtualization functions. These functions are performed by software/firmware executed by the NVD. Examples of network virtualization functions include without limitation: packet encapsulation and de-capsulation functions; functions for creating a VCN network; functions for implementing network policies such as VCN security list (firewall) functionality; functions that facilitate the routing and forwarding of packets to and from compute instances in a VCN; and the like. In certain embodiments, upon receiving a packet, an NVD is configured to execute a packet processing pipeline for processing the packet and determining how the packet is to be forwarded or routed. As part of this packet processing pipeline, the NVD may execute one or more virtual functions associated with the overlay network such as executing VNICs associated with compute instances in the VCN, executing a Virtual Router (VR) associated with the VCN, the encapsulation and decapsulation of packets to facilitate forwarding or routing in the virtual network, execution of certain gateways (e.g., the Local Peering Gateway), the implementation of Security Lists, Network Security Groups, network address translation (NAT) functionality (e.g., the translation of Public IP to Private IP on a host by host basis), throttling functions, and other functions.

**[0121]** In certain embodiments, the packet processing data path in an NVD may comprise multiple packet pipelines, each composed of a series of packet transformation stages. In certain implementations, upon receiving a packet, the packet is parsed and classified to a single pipeline. The packet is then processed in a linear fashion, one stage after another, until the packet is either dropped or sent out over an interface of the NVD. These stages provide basic functional packet processing building blocks (e.g., validating headers, enforcing throttle, inserting new Layer-2 headers, enforcing L4 firewall, VCN encapsulation/decapsulation, etc.) so that new pipelines can be constructed by composing existing stages, and new functionality can be added by creating new stages and inserting them into existing pipelines.

**[0122]** An NVD may perform both control plane and data plane functions corresponding to a control plane and a data plane of a VCN. Examples of a VCN Control Plane are also depicted in FIGs. 18, 19, 20, and 21 (see references 1816, 1916, 2016, and 2116) and described below. Examples of a VCN Data Plane are depicted in FIGs. 18, 19, 20, and 21 (see references 1818, 1918, 2018, and 2118) and described below. The control plane functions include functions used for configuring a network (e.g., setting up routes and route tables, configuring VNICs, etc.) that controls how data is to be forwarded. In certain embodiments, a VCN Control Plane is provided that computes all the overlay-to-substrate mappings centrally and publishes them to the NVDs and to the virtual network edge devices such as various gateways such as the DRG, the SGW, the IGW, etc. Firewall rules may also be published using the same mechanism. In certain embodiments, an NVD only gets the mappings that are relevant for that NVD. The data plane functions include functions for the actual routing/forwarding of a packet based upon configuration set up using control plane. A VCN data plane is implemented by encapsulating the customer's network packets before they traverse the substrate network. The encapsulation/decapsulation functionality is implemented on the NVDs. In certain embodiments, an NVD is configured to intercept all network packets in and out of host machines and perform network virtualization functions.

**[0123]** As indicated above, an NVD executes various virtualization functions including VNICs and VCN VRs. An NVD may execute VNICs associated with the compute instances hosted by one or more host machines connected to the VNIC. For example, as depicted in FIG. 2, NVD 210 executes the functionality for VNIC 276 that is associated with compute instance 268 hosted by host machine 202 connected to NVD 210. As another example, NVD 212 executes VNIC 280 that is associated with bare metal compute instance 272 hosted by host machine 206 and executes VNIC 284 that is associated with compute instance 274 hosted by host machine 208. A host machine may host compute instances belonging to different VCNs, which belong to different customers, and the NVD connected to the host machine may execute the VNICs (i.e., execute VNICs-relate functionality) corresponding to the compute instances.

**[0124]** An NVD also executes VCN Virtual Routers corresponding to the VCNs of the compute instances. For example, in the embodiment depicted in FIG. 2, NVD 210 executes VCN VR 277 corresponding to the VCN to which compute instance 268 belongs. NVD 212 executes one or more VCN VRs 283 corresponding to one or more VCNs to which compute instances hosted by host machines 206 and 208 belong. In certain embodiments, the VCN VR corresponding to that VCN is executed by all the NVDs connected to host machines that host

at least one compute instance belonging to that VCN. If a host machine hosts compute instances belonging to different VCNs, an NVD connected to that host machine may execute VCN VRs corresponding to those different VCNs.

**[0125]** In addition to VNICs and VCN VRs, an NVD may execute various software (e.g., daemons) and include one or more hardware components that facilitate the various network virtualization functions performed by the NVD. For purposes of simplicity, these various components are grouped together as "packet processing components" shown in FIG. 2. For example, NVD 210 comprises packet processing components 286 and NVD 212 comprises packet processing components 288. For example, the packet processing components for an NVD may include a packet processor that is configured to interact with the NVD's ports and hardware interfaces to monitor all packets received by and communicated using the NVD and store network information. The network information may, for example, include network flow information identifying different network flows handled by the NVD and per flow information (e.g., per flow statistics). In certain embodiments, network flows information may be stored on a per VNIC basis. The packet processor may perform packet-by-packet manipulations as well as implement stateful NAT and L4 firewall (FW). As another example, the packet processing components may include a replication agent that is configured to replicate information stored by the NVD to one or more different replication target stores. As yet another example, the packet processing components may include a logging agent that is configured to perform logging functions for the NVD. The packet processing components may also include software for monitoring the performance and health of the NVD and, also possibly of monitoring the state and health of other components connected to the NVD.

**[0126]** FIG. 1 shows the components of an example virtual or overlay network including a VCN, subnets within the VCN, compute instances deployed on subnets, VNICs associated with the compute instances, a VR for a VCN, and a set of gateways configured for the VCN. The overlay components depicted in FIG. 1 may be executed or hosted by one or more of the physical components depicted in FIG. 2. For example, the compute instances in a VCN may be executed or hosted by one or more host machines depicted in FIG. 2. For a compute instance hosted by a host machine, the VNIC associated with that compute instance is typically executed by an NVD connected to that host machine (i.e., the VNIC functionality is provided by the NVD connected to that host machine). The VCN VR function for a VCN is executed by all the NVDs that are connected to host machines hosting or executing the compute instances that are part of that VCN. The gateways associated with a VCN may be executed by one or more

different types of NVDs. For example, certain gateways may be executed by smartNICs, while others may be executed by one or more host machines or other implementations of NVDs.

**[0127]** As described above, a compute instance in a customer VCN may communicate with various different endpoints, where the endpoints can be within the same subnet as the source compute instance, in a different subnet but within the same VCN as the source compute instance, or with an endpoint that is outside the VCN of the source compute instance. These communications are facilitated using VNICs associated with the compute instances, the VCN VRs, and the gateways associated with the VCNs.

**[0128]** For communications between two compute instances on the same subnet in a VCN, the communication is facilitated using VNICs associated with the source and destination compute instances. The source and destination compute instances may be hosted by the same host machine or by different host machines. A packet originating from a source compute instance may be forwarded from a host machine hosting the source compute instance to an NVD connected to that host machine. On the NVD, the packet is processed using a packet processing pipeline, which can include execution of the VNIC associated with the source compute instance. Since the destination endpoint for the packet is within the same subnet, execution of the VNIC associated with the source compute instance results in the packet being forwarded to an NVD executing the VNIC associated with the destination compute instance, which then processes and forwards the packet to the destination compute instance. The VNICs associated with the source and destination compute instances may be executed on the same NVD (e.g., when both the source and destination compute instances are hosted by the same host machine) or on different NVDs (e.g., when the source and destination compute instances are hosted by different host machines connected to different NVDs). The VNICs may use routing/forwarding tables stored by the NVD to determine the next hop for the packet.

**[0129]** For a packet to be communicated from a compute instance in a subnet to an endpoint in a different subnet in the same VCN, the packet originating from the source compute instance is communicated from the host machine hosting the source compute instance to the NVD connected to that host machine. On the NVD, the packet is processed using a packet processing pipeline, which can include execution of one or more VNICs, and the VR associated with the VCN. For example, as part of the packet processing pipeline, the NVD executes or invokes functionality corresponding to the VNIC (also referred to as executes the VNIC) associated with source compute instance. The functionality performed by the VNIC may include looking at the VLAN tag on the packet. Since the packet's destination is outside the subnet, the VCN

VR functionality is next invoked and executed by the NVD. The VCN VR then routes the packet to the NVD executing the VNIC associated with the destination compute instance. The VNIC associated with the destination compute instance then processes the packet and forwards the packet to the destination compute instance. The VNICs associated with the source and destination compute instances may be executed on the same NVD (e.g., when both the source and destination compute instances are hosted by the same host machine) or on different NVDs (e.g., when the source and destination compute instances are hosted by different host machines connected to different NVDs).

**[0130]** If the destination for the packet is outside the VCN of the source compute instance, then the packet originating from the source compute instance is communicated from the host machine hosting the source compute instance to the NVD connected to that host machine. The NVD executes the VNIC associated with the source compute instance. Since the destination end point of the packet is outside the VCN, the packet is then processed by the VCN VR for that VCN. The NVD invokes the VCN VR functionality, which may result in the packet being forwarded to an NVD executing the appropriate gateway associated with the VCN. For example, if the destination is an endpoint within the customer's on-premise network, then the packet may be forwarded by the VCN VR to the NVD executing the DRG gateway configured for the VCN. The VCN VR may be executed on the same NVD as the NVD executing the VNIC associated with the source compute instance or by a different NVD. The gateway may be executed by an NVD, which may be a smartNIC, a host machine, or other NVD implementation. The packet is then processed by the gateway and forwarded to a next hop that facilitates communication of the packet to its intended destination endpoint. For example, in the embodiment depicted in FIG. 2, a packet originating from compute instance 268 may be communicated from host machine 202 to NVD 210 over link 220 (using NIC 232). On NVD 210, VNIC 276 is invoked since it is the VNIC associated with source compute instance 268. VNIC 276 is configured to examine the encapsulated information in the packet and determine a next hop for forwarding the packet with the goal of facilitating communication of the packet to its intended destination endpoint, and then forward the packet to the determined next hop.

**[0131]** A compute instance deployed on a VCN can communicate with various different endpoints. These endpoints may include endpoints that are hosted by CSPI 200 and endpoints outside CSPI 200. Endpoints hosted by CSPI 200 may include instances in the same VCN or other VCNs, which may be the customer's VCNs, or VCNs not belonging to the customer. Communications between endpoints hosted by CSPI 200 may be performed over physical

network 218. A compute instance may also communicate with endpoints that are not hosted by CSPI 200 or are outside CSPI 200. Examples of these endpoints include endpoints within a customer's on-premise network or data center, or public endpoints accessible over a public network such as the Internet. Communications with endpoints outside CSPI 200 may be performed over public networks (e.g., the Internet) (not shown in FIG. 2) or private networks (not shown in FIG. 2) using various communication protocols.

**[0132]** The architecture of CSPI 200 depicted in FIG. 2 is merely an example and is not intended to be limiting. Variations, alternatives, and modifications are possible in alternative embodiments. For example, in some implementations, CSPI 200 may have more or fewer systems or components than those shown in FIG. 2, may combine two or more systems, or may have a different configuration or arrangement of systems. The systems, subsystems, and other components depicted in FIG. 2 may be implemented in software (e.g., code, instructions, program) executed by one or more processing units (e.g., processors, cores) of the respective systems, using hardware, or combinations thereof. The software may be stored on a non-transitory storage medium (e.g., on a memory device).

**[0133]** FIG. 4 depicts connectivity between a host machine and an NVD for providing I/O virtualization for supporting multitenancy according to certain embodiments. As depicted in FIG. 4, host machine 402 executes a hypervisor 404 that provides a virtualized environment. Host machine 402 executes two virtual machine instances, VM1 406 belonging to customer/tenant #1 and VM2 408 belonging to customer/tenant #2. Host machine 402 comprises a physical NIC 410 that is connected to an NVD 412 via link 414. Each of the compute instances is attached to a VNIC that is executed by NVD 412. In the embodiment in FIG. 4, VM1 406 is attached to VNIC-VM1 420 and VM2 408 is attached to VNIC-VM2 422.

**[0134]** As shown in FIG. 4, NIC 410 comprises two logical NICs, logical NIC A 416 and logical NIC B 418. Each virtual machine is attached to and configured to work with its own logical NIC. For example, VM1 406 is attached to logical NIC A 416 and VM2 408 is attached to logical NIC B 418. Even though host machine 402 comprises only one physical NIC 410 that is shared by the multiple tenants, due to the logical NICs, each tenant's virtual machine believes they have their own host machine and NIC.

**[0135]** In certain embodiments, each logical NIC is assigned its own VLAN ID. Thus, a specific VLAN ID is assigned to logical NIC A 416 for Tenant #1 and a separate VLAN ID is assigned to logical NIC B 418 for Tenant #2. When a packet is communicated from VM1 406, a tag assigned to Tenant #1 is attached to the packet by the hypervisor and the packet is then

communicated from host machine 402 to NVD 412 over link 414. In a similar manner, when a packet is communicated from VM2 408, a tag assigned to Tenant #2 is attached to the packet by the hypervisor and the packet is then communicated from host machine 402 to NVD 412 over link 414. Accordingly, a packet 424 communicated from host machine 402 to NVD 412 has an associated tag 426 that identifies a specific tenant and associated VM. On the NVD, for a packet 424 received from host machine 402, the tag 426 associated with the packet is used to determine whether the packet is to be processed by VNIC-VM1 420 or by VNIC-VM2 422. The packet is then processed by the corresponding VNIC. The configuration depicted in FIG. 4 enables each tenant's compute instance to believe that they own their own host machine and NIC. The setup depicted in FIG. 4 provides for I/O virtualization for supporting multi-tenancy.

**[0136]** FIG. 5 depicts a simplified block diagram of a physical network 500 according to certain embodiments. The embodiment depicted in FIG. 5 is structured as a Clos network. A Clos network is a particular type of network topology designed to provide connection redundancy while maintaining high bisection bandwidth and maximum resource utilization. A Clos network is a type of non-blocking, multistage or multi-tiered switching network, where the number of stages or tiers can be two, three, four, five, etc. The embodiment depicted in FIG. 5 is a 3-tiered network comprising tiers 1, 2, and 3. The TOR switches 504 represent Tier-0 switches in the Clos network. One or more NVDs are connected to the TOR switches. Tier-0 switches are also referred to as edge devices of the physical network. The Tier-0 switches are connected to Tier-1 switches, which are also referred to as leaf switches. In the embodiment depicted in FIG. 5, a set of "n" Tier-0 TOR switches are connected to a set of "n" Tier-1 switches and together form a pod. Each Tier-0 switch in a pod is interconnected to all the Tier-1 switches in the pod, but there is no connectivity of switches between pods. In certain implementations, two pods are referred to as a block. Each block is served by or connected to a set of "n" Tier-2 switches (sometimes referred to as spine switches). There can be several blocks in the physical network topology. The Tier-2 switches are in turn connected to "n" Tier-3 switches (sometimes referred to as super-spine switches). Communication of packets over physical network 500 is typically performed using one or more Layer-3 communication protocols. Typically, all the layers of the physical network, except for the TORs layer are n-ways redundant thus allowing for high availability. Policies may be specified for pods and blocks to control the visibility of switches to each other in the physical network so as to enable scaling of the physical network.

**[0137]** A feature of a Clos network is that the maximum hop count to reach from one Tier-0 switch to another Tier-0 switch (or from an NVD connected to a Tier-0- switch to another NVD connected to a Tier-0 switch) is fixed. For example, in a 3-Tiered Clos network at most seven hops are needed for a packet to reach from one NVD to another NVD, where the source and target NVDs are connected to the leaf tier of the Clos network. Likewise, in a 4-tiered Clos network, at most nine hops are needed for a packet to reach from one NVD to another NVD, where the source and target NVDs are connected to the leaf tier of the Clos network. Thus, a Clos network architecture maintains consistent latency throughout the network, which is important for communication within and between data centers. A Clos topology scales horizontally and is cost effective. The bandwidth/throughput capacity of the network can be easily increased by adding more switches at the various tiers (e.g., more leaf and spine switches) and by increasing the number of links between the switches at adjacent tiers.

**[0138]** In certain embodiments, each resource within CSPI is assigned a unique identifier called a Cloud Identifier (CID). This identifier is included as part of the resource's information and can be used to manage the resource, for example, via a Console or through APIs. An example syntax for a CID is:

ocid1.<RESOURCE TYPE>.<REALM>.[REGION][.FUTURE USE].<UNIQUE ID>

where,

ocid1: The literal string indicating the version of the CID;

resource type: The type of resource (for example, instance, volume, VCN, subnet, user, group, and so on);

realm: The realm the resource is in. Example values are "c1" for the commercial realm, "c2" for the Government Cloud realm, or "c3" for the Federal Government Cloud realm, etc. Each realm may have its own domain name;

region: The region the resource is in. If the region is not applicable to the resource, this part might be blank;

future use: Reserved for future use.

unique ID: The unique portion of the ID. The format may vary depending on the type of resource or service.

**Multi-Cloud Introduction**

**[0139]** FIG. 6 depicts a simplified high-level diagram of a distributed environment 600 comprising multiple cloud environments provided by different cloud service providers (CSPs)

wherein the cloud environments include a particular cloud environment that provides specialized infrastructure that enables one or more cloud services provided by that particular cloud environment to be used by customers of other cloud environments according to certain embodiments. As depicted in FIG. 6, various different cloud environments (also referred to as "clouds") may be provided by different cloud service providers (CSPs), each cloud environment or cloud offering one or more cloud services that can be subscribed to by one or more customers of that cloud environment. The set of cloud services offered by a cloud environment provided by a CSP may include one or more different types of cloud services including but not restricted to Software-as-a-Service (SaaS) services, Infrastructure-as-a-Service (IaaS) services, Platform-as-a-Service (PaaS) services, Database-as-a-Service (DBaaS) services, and others. Examples of cloud environments provided by various CSPs include Oracle® Cloud Infrastructure (OCI) provided by Oracle Corporation, Microsoft® Azure provided by Microsoft Corporation, Google Cloud™ provided by Google LLC, Amazon Web Services (AWS®) provided by Amazon Corporation, and others. The cloud services offered by a particular cloud environment may be different from the set of cloud services offered by another cloud environment.

**[0140]** In a typical cloud environment, a CSP provides cloud service provider infrastructure (CSPI) that is used to provide the one or more cloud services that are offered by that cloud environment to its customers. The CSPI provided by a CSP may include various types of hardware and software resources including compute resources, memory resources, networking resources, consoles for accessing the cloud services, and others. A customer of a cloud environment provided by a CSP may subscribe to one or more of the cloud services offered by that cloud environment. Various subscription models may be offered by the CSP to its customers. After a customer subscribes to a cloud service provided by a cloud environment, one or more users may be associated with the subscribing customer and these users can use the cloud service subscribed to by the customer. In certain implementations, when a customer subscribes to a cloud service provided by a particular cloud environment, a customer account or customer tenancy is created for that customer. One or more users can then be associated with the customer tenancy and these users can then use the services subscribed to by the customer under the customer tenancy. Information regarding the services subscribed to by a customer, the users associated with the customer tenancy, etc., is usually stored within the cloud environment and associated with the customer tenancy.

**[0141]** For example, three different cloud environments provided by three different CSPs are depicted in FIG. 6. These include a Cloud Environment A (cloud A) 610 provided by a CSP A, a Cloud Environment B (cloud B) 640 provided by a CSP B, and a Cloud Environment C (cloud C) 660 provided by a CSP C. Cloud A 610 includes infrastructure CSPI_A 612 provided by CSP A, and this infrastructure may be used to provide a set of services "Services A" 614 offered by cloud A 610. One or more customers (e.g., Cust_A1 616-1, Cust_A2 616-2) may subscribe to one or more services from Services A 614 provided by cloud A 610. One or more users 618-1 may be associated with Customer A1 616-1 and can use the services subscribed to by customer A1 616-1 in cloud A 610. In a similar manner, one or more users 618-2 may be associated with customer A2 616-2 and can use the services subscribed to by customer A2 616-2 in cloud A 610. In various use cases, the services subscribed to by customer A1 616-1 may be different from the services subscribed to by customer A2 616-2.

**[0142]** As depicted in FIG. 6, cloud B 640 includes infrastructure CSPI_B 642 provided by CSP B, and this infrastructure may be used to provide a set of services "Services B" 644 offered by cloud B 640. One or more customers (e.g., Cust_B1 646-1) may subscribe to one or more services from Services B 644. One or more users 648-1 may be associated with Customer B1 646-1 and can use the services subscribed to by customer B1 646-1 in cloud B 640.

**[0143]** As depicted in FIG. 6, cloud C 660 includes infrastructure CSPI_C 662 provided by CSP C, and this infrastructure may be used to provide a set of services "Services C" 664 offered by cloud C 660. One or more customers (e.g., Cust_C1 666-1) may subscribe to one or more services from Services C 664. One or more users 668-1 may be associated with Customer C1 666-1 and can use the services subscribed to by customer C1 666-1 in cloud C 660. It is to be noted that Services A 614, Services B 644, and Services C 664 can be different from each other.

**[0144]** In existing cloud implementations, each cloud provides a closed ecosystem for its subscribing customers and associated users. As a result, a customer of a cloud environment and its associated users are restricted to using the services offered by the cloud that the customer subscribes to. For example, customer B1 646-1 and its users 648-1 are restricted to using services B 644 provided by cloud B 640 and cannot use their account in cloud B 640 to access services from a different cloud environment, such as a services from services A 614 offered by cloud A 610 or a service from Services C 664 offered by cloud C 660. The teachings described herein overcome this limitation. As described in this disclosure, various techniques are described that enable a link to be created between two cloud environments that enables a service provided by a first cloud environment provided by a first CSP to be used by a customer

(and associated users) of a second different cloud environment provided by a second different CSP, using the customer's account in the second cloud environment.

**[0145]** For example, in the embodiment depicted in FIG. 6, infrastructure CSPI_A 612 provided by CSP A, in addition to other infrastructure 620, includes special infrastructure 622 (referred to as multi-cloud enabling infrastructure 622 or MEI 622 or multi-cloud infrastructure 622) that enables one or more services 614 offered by cloud A to be used by customers and associated users of other clouds, such as clouds B 640 and C 660, using the customer accounts in those other clouds . In certain implementations, customers of clouds B and C do not have to open separate accounts with cloud A to use one or more of services 614 offered by cloud A 610. A customer B1 646-1 of cloud B 640 and an associated user 648-1 can use their customer account or tenancy in cloud B 640 to use one or more services 614 provided by cloud A 610. As another example, a customer C1 666-1 of cloud C 660 and an associated user 668-1 can use their customer account or tenancy in cloud C 660 to use one or more services 614 provided by cloud A 610.

**[0146]** In certain implementations, MEI 622 enables links to be created between cloud A 610 and other clouds, where these links can be used by customers of the other clouds and their associated users to access and make use of services provided by cloud A 610. This is symbolically shown in FIG. 6 as a link 670 created between cloud A 610 and cloud B 640, and a link 672 created between cloud A 610 and cloud C 660. Via link 670, a customer of cloud B 640 can access or use one or more services 614 provided by cloud A 610. Likewise, via link 672, a customer of cloud C 660 can access or use one or more services 614 provided by cloud A 610.

**[0147]** There are different ways in which MEI 612 may be implemented. In certain embodiments, MEI 612 may include components that enable links to be established with different clouds. For example, in FIG. 6, MEI 622 includes an infrastructure component 624 that is responsible for enabling link 670 with cloud B 640, and an infrastructure component 626 for enabling link 672 with cloud C 660. In a similar manner, MEI 622 may include other components that enable and facilitate links with other clouds. In some implementations, a component of MEI 622 may also facilitate links with multiple different clouds.

**[0148]** There are several reasons why a customer of one cloud may want or desire to use a cloud service provided by a different cloud. Using FIG. 6 as an example, there are multiple reasons why a customer B1 646-1 of cloud B 640 may want to use a cloud service 614 provided by cloud A 610. In one use case scenario, this may happen because cloud A 610 offers a cloud

service with functionality that is not provided by cloud B 640. As another use case scenario, clouds A and B may offer a similar service, but the service provided by cloud A 610 may be better (e.g., more features/functionality, faster, etc.) that the corresponding service offered by cloud B 640. As yet another use case scenario, customer B1 646-1 of cloud B 640 may want to use a cloud service provided by cloud A 610 because the service is provided at a cheaper price point than by cloud B 640. In some cases, there may be geographical restrictions or other reasons why customer B1 646-1 of cloud B 640 might want to use a cloud service provided by cloud A 610. For example, cloud A 610 may offer the desired service in a geographical area that is not serviced by cloud B 640 or the particular service is not provided by cloud B 640 in a geographical area in which the customer desires the service. Several other use case scenarios are also possible as to why a customer of one cloud might want to use a service provided by a different cloud.

**[0149]** In certain embodiments, MEI 622 provides capabilities and performs functions for creating the link between cloud A 610 and another cloud, and via the link, enabling a user associated with a customer of the other cloud to, in a seamless manner, access and use, from the other cloud itself, a service provided by cloud A 610. For example, MEI 622 enables a user 648-1 associated with customer B1 646-1 of cloud 640 to access a service from services A 614 provided by cloud A 610 in a seamless manner. In certain implementations, user interfaces (e.g., a console) may be provided that user 648-1 can access from within cloud B 640 that enable the user to see a list of services 614 offered by cloud A 610 and to select a particular service that the user 648-1 desires to access. In response to the user selection, MEI 622 is responsible for performing processing that establishes link 670 between clouds A and B to enable access to the requested service. The processing for setting up link 670 is performed substantially automatically by MEI 622. Customer B1 646-1 or associated users 648-1 do not have worry about performing any system, networking, or other configuration changes that are needed to facilitate the creation, maintenance, and usage of link 670 between clouds A 610 and B 640. No burden is placed on the users or the customers in the creation of the link between the clouds. The link is created in a fast and efficient manner using the techniques described in this disclosure.

**[0150]** MEI 622 may use various techniques to make the creation and use of the link seamless to users and customers and thus provide for an enhanced user experience. In certain implementations, MEI 622 causes the user interfaces (e.g., graphical user interfaces GUIs, etc.) and process flows that a customer B1 and associated users 648-1 interact with, such as for

requesting a service from cloud A 610 and for accessing the requested service from cloud A 610, to be substantially similar to the interfaces and process flows that the customer/user would experience in cloud B 640. In this manner, the customer or user, who may be accustomed to the interfaces and process flows of cloud B 640, does not have to learn new interfaces and process flows to access a service 614 from cloud A 610. MEI 622 may present different interfaces and process flows for users of different cloud environments. For example, a first set of user interfaces and process flows that are substantially similar to the user interfaces and flows of cloud B may be presented to a user from cloud B 640, while a different set of user interfaces and process flows that are substantially similar to the user interfaces and flows of cloud C may be presented to a user accessing cloud A 610 from cloud C 660. This is done to simplify and consequently enhance a user's experience for accessing services 614 of cloud A 610 from other clouds.

**[0151]** As another example, each cloud environment typically includes an identify management system that is configured to provide security for the cloud environment. The identity management system is configured to protect resources in the cloud environment, including resources provided by the CSP and resources of subscribing cloud customers that are deployed in the cloud environment. Functions performed by the identity management system include, for example, managing identity credentials (e.g., usernames, passwords, etc.) associated with the cloud's subscribing customers and associated users, using the identity credentials to regulate users' access to cloud resources and services based upon permission/access policies configured for the cloud environment, and other functions. Different clouds may use different identity management systems and associated techniques. For example, the identity management system and associated procedures in cloud A 610 may be completely different from the identity management system and associated procedures in cloud B 640, which in turn may be completely different from the identity management system and associated procedures in cloud C 660. In certain implementations, in spite of these differences in identity management systems and associated procedures between different cloud environments, the techniques described herein enable a user associated with a customer of a first cloud to access a cloud service provided by a different cloud using the same identity credentials associated with the customer and the user in the first cloud.

**[0152]** For example, in the embodiment depicted in FIG. 6, cloud B 640 provided by CSP B may include an identity management system that assigns or allocates identity credentials to its subscribing customers and associated users, such as customer B1 646-1 and associated users

648-1. These identity credentials are associated with the tenancy created for customer B1 646-1 in cloud B 640. In certain implementations, MEI 622 provided by cloud A 610 enables a user 648-1 associated with cloud B customer B1 646-1 to access a service from services A 614 in cloud A 610 using identity credentials associated with users 648-1 and customer B1 646-1 in cloud B 640. This greatly enhances the user experience for users 648-1 since they do not have to create new identity credentials that are specific to cloud A 610 just for the purpose of accessing a service 614 in cloud A 610. The MEI 622 facilitates such access.

**[0153]** As an example, a customer B1 of cloud B 640 may select to use a service, such as a Database-as-a Service (DBaaS), from the set of services 614 provided by cloud A 610. In response to such a selection, MEI 622 causes a link 670 to be automatically created between cloud A 610 and cloud B 640 to enable users 648-1 associated with customer B1 646-1 to use the DBaaS service provided by cloud A 610. The automatic setup of link 670 is facilitated by MEI 622. After link 670 has been set up, a user 648-1 can use the DBaaS service in cloud A 610 via cloud B 640. As part of using this service, user 648-1 can, via cloud B 640 send a request to cloud A 610 to create a database resource. In response, CSPI_A 612 may create the requested database in cloud A 610. In certain implementations, the created database may be provisioned in a virtual network (e.g., a virtual cloud network or VCN) created for customer B1 in cloud A 610 and is accessible to user 648-1 via cloud B 640. User 648-1 may then send, from cloud B 640, requests to cloud A 610 to use the provisioned database. These requests may include, for example, requests to write data to the database, to update data stored in the database, to delete data in the database, to delete the database, to create additional databases, and the like. In some use cases, these requests may originate from a user 648-1 via cloud B 640 or from a service 644 provided by cloud B 640. In this manner, MEI 622 provided by cloud A 610 enables a user associated with a customer of a different cloud provided by a different CSP to seamlessly access a service provided by cloud A 610.

**[0154]** Distributed environment 600 depicted in FIG. 6 is merely an example and is not intended to unduly limit the scope of claimed embodiments. Many variations, alternatives, and modifications are possible. For example, in alternative embodiments, distributed environment 600 may have more or fewer cloud environments. The cloud environments may also have more or fewer systems and components or may have a different configuration or arrangement of the systems and components. The systems and components depicted in FIG. 6 may be implemented in software (e.g., code, instructions, program) executed by one or more processing units (e.g., processors, cores) of the respective systems, using hardware, or

combinations thereof. The software may be stored on a non-transitory storage medium (e.g., on a memory device).

**Multi Cloud Control Plane (MCCP)**

**[0155]** FIG. 7 depicts a high-level architecture of a Multi-Cloud Control Plane (MCCP) according to some embodiments. As shown in FIG. 7, the high-level architecture 700 includes a first cloud environment provided by a first cloud services provider (e.g., OCI) 720 and a second cloud environment provided by a second cloud services provider 710 (i.e., Azure). The first cloud environment 720 includes a first cloud infrastructure 720A that provides users of the first cloud environment 720 with a plurality of services. Additionally, the first cloud environment 720 includes a multi-cloud infrastructure 720B that provisions for capabilities to deliver services of the first cloud infrastructure 720A (e.g., Oracle Cloud Infrastructure (OCI)) to users of other cloud environment (e.g., the second cloud infrastructure 710A). The multi-cloud infrastructure 720B allows users to access services (e.g., Oracle PaaS services) on other clouds with a user experience as close as possible to that of the native cloud environments of the users, while providing simple integration between the cloud environments.

**[0156]** The second cloud infrastructure 710A includes a second cloud portal 711, an active directory 712, a resource manager 713, a customer subscription 715, a subscription of the first cloud provider 717. The second cloud portal 711 is a centralized access point where customers of the second environment 710 can login and manage their cloud deployments and instances. It is noted that the second cloud portal may provide options for both monitoring and operating services provided by the second cloud infrastructure. The active directory 712 is a service provided by the second cloud infrastructure 710A that that provides administrators with the ability to manage end-user identities and access privileges. Its services may include core directory, access management, and identity protection. The resource manager 713 is a deployment and management service of the second cloud infrastructure i.e., a management layer that provides users to perform operations (e.g., create, update, delete, etc.,) with respect to resources deployed in the customer subscription 715. It is noted that the customer subscription may also be referred to as a virtual network (VNET) where customer applications are deployed and executed. The subscription of the first cloud provider 717 in the second cloud infrastructure 710A includes an express route and a hub and spoke VNET that provision for network connections (e.g., from on-premises locations, from external cloud environments) to be established with the second cloud infrastructure 710A. It is noted that such connections may

not be routed through the public internet, thereby providing users with more reliability, faster speeds, consistent latency, and higher security.

**[0157]** The first cloud infrastructure 720A includes a control plane 724, a customer tenancy 726, and a multi-cloud infrastructure 720B. The control plane 724 of the first cloud infrastructure 720A is a native control plane of the first cloud environment that provides management and orchestration across the cloud environment. It is here where configuration baselines are set, user and role access provisioned, and applications reside so they can be executed with related services. The multi-cloud infrastructure 720B includes a multi-cloud platform data plane 722 and multi-cloud platform data plane 728. As stated previously, the multi-cloud infrastructure 720B provisions for users of other cloud environments (e.g., the second cloud environment 710) to access services provided by the first cloud environment with a user experience as close as possible to that of the native cloud environments of the users (e.g., the second cloud environment 710), while providing simple integration between the cloud environments.

**[0158]** The MCCP architecture of FIG. 7 further includes a multi-cloud console 721 (different than the second cloud portal 711) that permits users authenticated in the second cloud infrastructure 710 to perform control plane operations on resources of the first cloud infrastructure 720 that are exposed via the multi-cloud infrastructure 720B. In some implementations, as shown in FIG. 7, all user 705 requests transmitted to the multi-cloud console 721 are directed to the multi-cloud infrastructure included in the first cloud environment. It is appreciated that the user 705 can transmit requests (e.g., CRUD requests) with respect to resources provided by the first cloud infrastructure directly to the multi-cloud console 721. The multi-cloud console 721 is configured to process requests that have a syntax or format similar to that as used locally in the second cloud infrastructure. In other words, the multi-cloud console 721 has a similar look-and-feel as well as uses terminology that is similar to the second cloud portal 711 included in the second cloud infrastructure 710A. In some implementations, a link (e.g., a weblink) may be provided in the second cloud console 711 that directs users to the multi-cloud console 721.

**[0159]** The multi-cloud infrastructure 720B included in the first cloud infrastructure 720A includes a plurality of microservices such as an authority module 722A, a proxy module 722B, a platform services module 722C, a cloud-link adaptor 722D, a pool of adaptors 722E including adaptor 1, adaptor 2, adaptor 3, and adaptor 4, and a network link adaptor 722F. The pool of adaptors 722E can include adaptors such as an Exa-data cloud service adaptor, an autonomous

database-shared adaptor, an autonomous database-dedicated adaptor, and a virtual machine database adaptor.

**[0160]** Each of the adaptors included in the pool of adaptors 722E is responsible for exposing a set of unique underlying resources (provided by the first cloud infrastructure 720A) to users of other cloud environments (e.g., second cloud environment). Specifically, each of the adaptors in the pool of adaptors 722E maps to a particular product or resource offered by the first cloud infrastructure 720A. It is noted that the actual resources are created by the native control plane 724 of the first cloud infrastructure. For instance, with respect to database as a service (DBaaS), the DBaaS control plane included in the control plane 724 is configured to instantiate Exa-database resources in the customer tenancy 726 of the first cloud environment.

**[0161]** The incoming request received by the multi-cloud infrastructure 720B is processed by the authority module 722A for performing authentication and access control. Each request includes a token associated with the account of the user in the second cloud infrastructure. The authority module extracts the token and validates the token in conjunction with the active directory 712 (i.e., the identity provider system of the second cloud infrastructure 710A). Upon successful validation, the authority module 722A may check roles (i.e., set of privileges) associated with the user. It is noted that a role may be associated with one or more tasks/operations that are permitted for the role. According to one embodiment, the authority module 722A is responsible for authenticating incoming requests to MCCP and authorizing if the user is allowed to perform the requested operation based on the roles associated with the token. In some implementations, the authority module 722A may perform the authentication process described above by taking advantage of a custom authentication feature of a service platform (i.e., SPLAT associated with the first cloud infrastructure). SPLAT accepts an incoming request and forwards it to the authority module 722A, which further parse the incoming request to determine an authorization decision and returns a success or failure message back to SPLAT. On success, the request is passed by SPLAT to the routing proxy 722B, whereas on failure, SPLAT returns an error response directly to the caller.

**[0162]** The proxy module 722B (also referred to as a routing proxy module) included in the multi-cloud infrastructure 720B is a component that receives an incoming request from the multi-cloud console 721 and routes the request to a particular adaptor included in the pool of adaptors 722E. By one embodiment, the proxy module 722B accepts pre-authenticated requests from a service platform (i.e., SPLAT) of the first cloud infrastructure and routes the requests to the appropriate adaptor based on path information included in the incoming request. In some

implementations, the proxy module 722B extracts an identifier (from the incoming request) corresponding to a provider of the service and routes the request to the appropriate adaptor in the pool of adaptors 722E. For example, referring to an exemplary request depicted in FIG. 10, it is noted that the service 1010 corresponds to an Exa-database infrastructure service. Thus, such a request is forwarded to a database adaptor included in the pool of adaptors 722E.

**[0163]** The cloud-link adaptor 722D included in the multi-cloud infrastructure 720B is responsible for handling lifecycle operations of resources provided by the first cloud infrastructure. The cloud-link adaptor 722D is configured to create a mapping (or a relationship created at a sign-up process) between an active directory tenant of the second cloud infrastructure (and its associated subscriptions) and a corresponding tenancy/account of the user in the first cloud infrastructure. In other words, the cloud-link adaptor 722D generates a mapping of a first identifier associated with the tenancy of the user in the first cloud infrastructure to a second identifier associated with the account of the user in the second cloud infrastructure.

**[0164]** In some implementations, the cloud-link adaptor 722D performs translation between external cloud identifiers (e.g., second identifier associated with the account of the user in the second cloud infrastructure) and a first identifier (associated with the tenancy of the user in the first cloud infrastructure) to enable operations going through the multi-cloud control plane 722 to map to the appropriate underlying resource in the first cloud infrastructure. In some embodiments, the cloud-link adaptor generates a data object to store the above-described mapping information. Additionally, the cloud-link adaptor 722D also generates a resource-principal that is associated with the data object. The resource-principal is assigned one or more permissions based on the token (and associated roles thereof) included in the request. Access to downstream services provided by the first cloud infrastructure is achieved by the user from the second cloud infrastructure based on the resource-principal. The cloud-link adaptor 722D may store the data object and the associated resource-principal in a root compartment of a tenancy of the user in the first cloud infrastructure. Alternatively, or additionally, the cloud-link adaptor 722D may also locally persist the data object and the resource-principal on the platform services module 722C of the multi-cloud infrastructure for seamless access by other adaptors included in the multi-cloud infrastructure.

**[0165]** The network link module (also referred to as a network adaptor) 722F is responsible for creating a network link between the customer subscription 715 (in the second cloud infrastructure) and the corresponding customer tenancy/account (in the first cloud

infrastructure) 726. By some embodiments, the network link module 722F obtains a token (from the platform services module 722C) and creates (1) a first peering relationship (in the first cloud environment) between the multi-cloud platform data plane 728 and the customer tenancy 726, and (2) a second peering relationship (in the second cloud environment) between the customer subscription 715 and the subscription of the first cloud services provider 717 included in the second cloud infrastructure.

**[0166]** The network link module 722F is also configured to establish network connectivity between the first cloud environment and the second cloud environment i.e., the network link module 722F can configure an interconnect 719 to communicatively couple the two cloud environments. It is appreciated that upon forming the network link between the two cloud environments, applications that are executed in the customer's subscription (e.g., in a VNET of the second cloud infrastructure) are able to access resources e.g., Exa-database resource that is deployed in the customer tenancy 726 of the first cloud infrastructure. Further, as shown in FIG. 7, within the tenancy 726, there is provided a functionality of telemetry. Such a functionality relates to maintaining logs, metrics, and other performance parameters related to the resources deployed in the customer tenancy and their respective usages. According to some embodiments, MCCP mirrors the logs and metrics associated with different resources in the first cloud environment and publishes the metrics, logs, events etc., for instance in a dashboard (e.g., in an application insights module included in the customer subscription 715 of the second cloud environment) for further processing.

**[0167]** By some embodiments, the platform services module 722C included in multi-cloud infrastructure is configured to store credentials associated with services of the first cloud infrastructure provided to the second cloud infrastructure. The platform services module 722C provides, for instance, tokens/resource principals for the different adaptors included in the pool of adaptors 722E so that the adaptors can communicate with native control plane 724 of the first cloud infrastructure. By some embodiments, the platform services module 722C exposes APIs that are called by different adaptors to perform tasks such as:

- Vending a minimally scoped access token (issued by the second cloud infrastructure) to adaptors. For example, the network adaptor 722F requires an access token to perform the above-described network peering operations.
- Providing a resource principal that adaptors will use to call downstream services to create resources in customer tenancies of the first cloud infrastructure.

- Triggering replication of observability data (logs, metrics, events) from the first cloud infrastructure to the second cloud infrastructure.

**[0168]** As stated previously, the pool of adaptors includes a plurality of adaptors, each of which is responsible for exposing a set of unique underlying resources of the first cloud infrastructure to the users of the second cloud infrastructure i.e., each adaptor maps to a particular product or resource offered by the first cloud environment. For instance, the Exa-database adaptor acts as a proxy for the users of the second cloud infrastructure to create and utilize Exa-database resources. Exa-database is a pre-configured combination of hardware and software that provides an infrastructure for executing databases. By some embodiments, Exa-database comprises a stack of resources: (a) Exadata infrastructure (i.e., hardware), (b) VM cloud cluster, (c) container databases, and (d) pluggable databases. According to some embodiments, the multi-cloud infrastructure provides the ability (for users of the second cloud infrastructure) to analyze each of the levels of stacked infrastructure. Moreover, the MCCP provides flexibility for a user to simply issue a create command (via the multi-cloud console 721) for a workflow, where after the MCCP performs automatic creation of individual resources at each level of the stack. It is appreciated that although the pool of adaptors 722F as depicted in FIG. 7 includes four different adaptors, it is in no way limiting the scope of the MCCP architecture 700. The MCCP architecture may include other adaptors, for example, a dedicated adaptor directed for use by a specific cloud service provider based on requirements of the cloud services provider.

**[0169]** FIGs. 8A and 8B depict exemplary flow diagrams for linking two user accounts in different cloud environments, according to some embodiments. The two user accounts may correspond to a first user account (e.g., tenancy) in a first cloud infrastructure and a second user account in a second cloud infrastructure. FIG. 8A depicts the process of linking the two user accounts, wherein the tenancy of the user is initially created in the first cloud infrastructure and then linked to the account of the user in the second cloud infrastructure. FIG. 8B depicts the process of linking the two user accounts, wherein the tenancy of the user in the first cloud infrastructure is already created i.e., the tenancy already exists.

**[0170]** FIG. 8A depicts a data flow that is executed when a user representing a customer, such as a system administrator makes a call from the second cloud infrastructure to the multi-cloud infrastructure (included in the first cloud infrastructure) for signing up for services provided in the first cloud infrastructure. A special console (referred to herein as a multi-cloud console) may be provided for users of the second cloud infrastructure to open accounts within

the first cloud infrastructure and to link the account of the user in the first cloud infrastructure to the account of the user in the second cloud infrastructure. It is noted that linking the accounts in the first and second cloud infrastructures, enables the user to utilize (from the second cloud infrastructure), one or more services provided by the first cloud infrastructure. In certain implementations, the user signs up for the services using the multi-cloud console. In response to the sign-up, a URL may be sent to the user that enables the user to log into the multi-cloud console. The multi-cloud console exposes UIs and APIs that are similar to the UIs and APIs of the second cloud infrastructure.

**[0171]** The multi-cloud console may provide various UIs (e.g., GUIs) that provide user-selectable options that enable a user of the second cloud infrastructure to open an account in the first cloud infrastructure and to further link the user accounts in the first and second cloud infrastructures. For example, multi-cloud console may provide a sign-up UI, which enables a user to create an account/tenancy in the first cloud infrastructure and to link the user's account (in the first cloud infrastructure) to the user's account in the second cloud infrastructure. Once the processing triggered responsive to the signup/link request is complete the user's accounts in the respective cloud infrastructures are linked together. As part of the processing, the multi-cloud console enables a user (e.g., a system administrator) to log into the second cloud infrastructure, and request: (a) creation of an account for the user (in the first cloud infrastructure) and to link the user's accounts; or (b) for an account that the user already has in the first cloud infrastructure, to request the user's first cloud infrastructure account to be linked to the user's second cloud infrastructure account.

**[0172]** Accordingly, in certain use cases, there are two possibilities: (a) the user already has an existing account or tenancy in the first cloud infrastructure and now, via the multi-cloud console sign-up UI, the user requests a link to be created between the user's accounts (details pertaining to this processing is described next with reference to FIG. 8B), or (b) the user requests both the creation of a new account in the first cloud infrastructure and the linking of the newly created account with the user's account in the second cloud infrastructure. Details pertaining to this processing is described next with reference to FIG. 8A. It is appreciated that linking of the accounts enables a user to, via the second cloud, use resources in the first cloud. For example, via the second cloud, a user can request a resource to be created or provisioned in the first cloud, utilize and manage resources in the first cloud, delete resources in the first cloud, and the like.

**[0173]** As shown in FIG. 8A, in step 1, in response to a user's input provided to multi-cloud console (e.g., sign-up UI of the multi-cloud console) requesting a new account for the user to be created in the first cloud infrastructure and linked to the user's account in the second cloud infrastructure, a call is made by the multi-cloud console to accounts services (in the first cloud infrastructure) to create a new account. It is noted that the call includes a token that is generated by second cloud infrastructure. The token generated by the second cloud infrastructure and included in the call to the accounts services enables the account or tenancy created in the first cloud infrastructure to be linked to the user's account in the second cloud infrastructure.

**[0174]** At step 2, as part of the processing for setting up the new account for the user, accounts service may make a call to the authority/proxy module included in the multi-cloud control plane (e.g., authority module 722A included in the multi-cloud infrastructure of FIG. 7) to validate the token. The authority/proxy module validates the token (as described previously with reference to FIG. 7) and further processing may continue only upon successful validation of the token. Upon successful validation of the token, accounts service creates a new account for the user in the first cloud infrastructure. As part of creating the new account, a native user may be created and associated with the newly created account. However, note that this native user's credentials are not used. Rather, the user's identity in the second cloud infrastructure is used for managing the account and its resources (from the second cloud infrastructure) in the first cloud infrastructure.

**[0175]** After the new account has been successfully created by accounts services in the first cloud infrastructure, at step 3, accounts services makes a call to the cloud-link adaptor (included in the multi-cloud infrastructure) to create a link (also referred to as a cloud-link) between the user's account in the second cloud infrastructure and the newly created account in the first cloud infrastructure. In certain implementations, the cloud-link adaptor exposes APIs to account services and to multi-cloud console. These APIs may be called by account services (or by the multi-cloud console) to request linking of the accounts. It is noted that in some implementations, there is no token included in the call of step 3, since the call requesting the linkage is not made by the user but is rather initiated by accounts services. In this case, cloud-link adaptor does not do another authentication that would require another token because it relies upon accounts services having already performed the necessary authentication for the user as part of the processing for setting up the account using the token received in step 1.

**[0176]** In some implementations, the processing performed in step 3 by the cloud-link adaptor for linking the two accounts comprises:

(a) cloud-link adaptor creates a data object (also referred to herein as a cloud-link resource object for storing metadata information identifying the two accounts being linked. For example, the data object stores metadata information including a mapping of a first identifier associated with the tenancy (i.e., account) created in the first cloud infrastructure and a second identifier associated with the account of the user with the second cloud service provider.

(b) cloud-link adaptor creates a new compartment for storing and containing resources on the first cloud infrastructure side that the user can/would manage using the multi-cloud console. In some implementations, the cloud-link resource object is created at the root of the customer's account in the first cloud infrastructure.

(c) cloud-link adaptor creates a new resource-principal (referred to herein as a cloud-link resource principal) for a resource that is desired to be used by the user of the second cloud infrastructure.

(d) cloud-link adaptor may perform one or more federation setups that facilitate linking of a domain in the first cloud infrastructure to an active directory (e.g., Active Directory 712 of the second cloud infrastructure as shown in FIG. 7). The process of federating accounts of the first cloud infrastructure and the second cloud infrastructure allows users/groups of users in the second cloud infrastructure to be authenticated in first cloud infrastructure, and users/groups from the first cloud infrastructure to be authenticated into the second cloud infrastructure.

**[0177]** In certain implementations, permissions are associated with the cloud-link resource principal based upon the user's credentials/token in the account of the second cloud infrastructure. Consent for setting up the resource principal is provided by the user when the user requests a new account using the multi-cloud console or requests the user's account in the first cloud infrastructure to be linked to the user's account in the second cloud infrastructure. Further, as shown in step 4, the cloud-link resource principal is transmitted to the downstream service to enable the user to utilize the downstream service(s) (e.g., one or more services provided by the first cloud infrastructure) from the second cloud infrastructure.

**[0178]** Turning to FIG. 8B, there is depicted processing related to linking an existing account or tenancy in the first cloud infrastructure to an account of the user in the second cloud infrastructure. As shown in FIG. 8B, the user, via the sign-up UI provided by the multi-cloud console, requests the user's accounts in the first cloud infrastructure and the second cloud infrastructure to be linked. In response, in step 1, the multi-cloud console makes a call to the cloud-link adaptor to link the accounts. Such a call may be made using one of the APIs exposed

by the cloud-link adaptor to the sign-up UI. In certain implementations, the information passed to the cloud-link adaptor in step 1, includes information identifying the user's account in the second cloud infrastructure and also the user's account in the first cloud infrastructure, a token issued by the second cloud infrastructure, etc.

**[0179]** At step 2, the cloud-link adaptor makes a call to the authority/proxy module included in the multi-cloud infrastructure to authenticate the token received in step 1. As part of this authentication, the authority/proxy module determines whether the user making the request has sufficient authority on the second cloud infrastructure side to make the linkage request. As part of the validation, roles and permissions set up on the second cloud infrastructure side may be checked. In response to the user being successfully validated, the cloud-link adaptor retrieves from the data object previously created for the user, the resource-principal associated with a resource that is desired to be utilized by the user. Further, as shown in step 3, the cloud-link resource principal is transmitted to the downstream service to enable the user to utilize the downstream service(s) (e.g., one or more services provided by the first cloud infrastructure) from the second cloud infrastructure.

**[0180]** FIG. 9 depicts an exemplary system diagram illustrating components of a multi-cloud control plane (MCCP) according to some embodiments. The MCCP 900 includes a service platform (SPLAT) 910, a routing proxy 915, a cloud-link adaptor 920, a database adaptor 925, a network adaptor 930, and an MCCP platform 935. Upon the user successfully completing the sign-up process as described above with reference to FIGs. 8A and 8B, the user may utilize the multi-cloud console 905 to issue commands to access, create, or update a resource in the tenancy of the user in the first cloud infrastructure. For sake of illustration, in what follows, there is described a scenario of the user utilizing the multi-cloud console to issue a request to create an Exa-database resource.

**[0181]** In some implementations, the user accesses the multi-cloud console 905 and provides login information e.g., credentials of the user in the second cloud infrastructure. The multi-cloud console 905 provides a plurality of options e.g., create a resource, access a resource, update a resource etc. Such options may be provided to the user in the form of selectable icons (e.g., buttons) in the multi-cloud console 905. Upon the user performing a selection (e.g., to create a resource), an API call is triggered to the service platform 910. It is appreciated that the request made to the service platform 910 is not a native call with respect to the first cloud infrastructure. Rather, the call is a REST type call including an authorization header that comprises a token associated with the user in the second cloud infrastructure.

**[0182]** The REST call including the token is further forwarded to the routing proxy module 915 that performs authentication and access control operations. According to some embodiments, the routing proxy module 915 performs an authentication operation by extracting the token included in the REST call. The routing proxy module 915 validates the token by comparing a signature (used to sign the request) with a publicly available signature of the second cloud infrastructure to ensure that the request originates from a valid customer associated with the second cloud infrastructure. Additionally, the routing proxy module 915 may also check roles i.e., privileges associated with the token e.g., whether the role corresponds to an Exadata DB administrator or the like. Based on the role, the routing proxy module 915 may route the request to an appropriate adaptor included in the MCCP framework 900.

**[0183]** According to one embodiment, the routing proxy module 915 compares the role (associated with the token) to a preconfigured list of roles that is published and assigned (as part of the API specification) for each of the adaptors. For example, if the role associated with the token corresponds to an 'Exadata DB administrator', then the request may be comprehended as one being of creating an Exa-database and thus the request is forwarded to the database adaptor 925. Additionally, by some embodiments, the routing proxy module 915 may analyze information included in the REST call such as a provider ID, resource type requested, etc., and based on the analyzed information, the routing proxy module 915 may forward the request to the appropriate adaptor. In some implementations, the routing proxy module 915 may route the request to an appropriate adaptor based on the API call being made by the user e.g., an API call for creating a database may be routed to a first adaptor in the pool of adaptors, as opposed to an API call for accessing a particular resource that may be routed to a different adaptor in the pool of adaptors. It is however noted that the routing proxy module 915 (prior to routing a particular request to an adaptor) checks the role(s) associated with a token that is included in the request, to verify whether the user has sufficient privileges to issue the request.

**[0184]** In some implementations, the request obtained by the routing proxy module 915 may not include information identifying the tenancy of the user in the first cloud infrastructure where the resource is to be deployed. Thus, the routing proxy module 915 communicates with the cloud-link adaptor 920 to obtain mapping information of the user's account in the second cloud infrastructure to the tenancy of the user in the first cloud infrastructure. If the mapping information exists, then the routing proxy module 915 obtains the information pertaining to the tenancy of the user in the first cloud infrastructure and passes the information to the database

adaptor 925. In this manner, the database adaptor 925 is aware of the tenancy of the user in the first cloud infrastructure where the resource is to be created/deployed. However, if the cloud-link adaptor 920 determines that no mapping information exists, then the routing proxy module 915 may simply issue, as a response to the request to create the database resource, an 'unauthorized-access' message that is transmitted back to the user.

**[0185]** It is noted that in some implementations, the cloud-link adaptor 920 creates a data object (referred to herein as a cloud-link resource object) for storing metadata information identifying the two accounts being linked. For example, the data object stores metadata information including a mapping of a first identifier associated with the tenancy (i.e., account) in the first cloud infrastructure and a second identifier associated with the account of the user with the second cloud service provider. Additionally, the cloud-link adaptor 920 also creates a resource-principal (referred to herein as a cloud-link resource principal) for a resource (e.g., database) that is desired to be created/managed by the user of the second cloud infrastructure. The cloud-link adaptor 920 may maintain the data object as well as the resource-principal within a root compartment of the tenancy of the user in the first cloud infrastructure. In some embodiments, the cloud-link adaptor 920 may also locally persist the data object and/or the resource principal in the MCCP platform 935.

**[0186]** In some embodiments, the database adaptor 925 may communicate with (or instruct) the network adaptor 930 to create a network link between the user's account in the second cloud infrastructure and the tenancy of the user in the first cloud infrastructure. For instance, the network adaptor 930 may obtain from the native services in the second cloud infrastructure module 945, a token associated with the user and create: (1) a first peering relationship (in the first cloud environment) between a data plane of the MCCP and the tenancy of the user in the first cloud infrastructure, and (2) a second peering relationship (in the second cloud environment) between the user's account and a subscription of the first cloud provider included in the second cloud infrastructure.

**[0187]** The network adaptor 930 is also configured to establish network connectivity between the first cloud infrastructure and the second cloud infrastructure i.e., the network adaptor 930 can configure an interconnect (e.g., interconnect 719 of FIG. 7) that communicatively couples the two cloud environments. It is appreciated that upon forming the network link between the tenancy of the user in the first cloud infrastructure and the account of the user in the second cloud infrastructure, applications that are executed in the user's subscription/account are able to access resources e.g., Exa-database deployed in the tenancy of

the first cloud infrastructure. Furthermore, the creation of the peering relationships provisions for metrics e.g., database usage metrics to be accessible in the second cloud infrastructure e.g., in a dashboard application that is executed in the subscription of the user in the second cloud infrastructure.

**[0188]** In some implementations, the database adaptor 925 may obtain the resource principal that is locally persisted in the MCCP platform 935. The database adaptor 925 may transmit a request (including the resource-principal) to one or more downstream services included in the first cloud infrastructure 940 to create the resources in the tenancy of the user in the first cloud infrastructure. In other words, the downstream services included in the first cloud infrastructure 940 utilizes the identity i.e., resource principal obtained from the MCCP platform 935 to create/deploy the required resources e.g., Exa-database in the tenancy of the user in the first cloud infrastructure. Upon the user issuing the request to create the Exa-database, the user may intermittently poll the MCCP 900 to obtain a status of the request. Upon the downstream services of the first cloud infrastructure 940 creating the resource in the tenancy of the user in the first cloud infrastructure, and the network adaptor 930 establishing the peering relationships, the MCCP 900 may notify the user regarding a successful completion of the request.

**[0189]** FIG. 10 depicts an exemplary request received by a multi-cloud infrastructure, according to some embodiments. Specifically, FIG. 10 depicts a request 1000 that received by the multi-cloud infrastructure from the multi-cloud console. The request 1000 includes metadata information required by the multi-cloud infrastructure to create a resource in a tenancy of the user in the first cloud infrastructure. The metadata information may include: (a) information identifying a subscription (e.g., an account) of the user (as well as a resource group information) in the second cloud infrastructure 1005, (b) information identifying a provider of the service 1010, and (c) a type of resource 1015 that is desired to be utilized by the user of the second cloud infrastructure.

**[0190]** In some implementations, the request 1000 is received by the proxy module (e.g., proxy 722B of FIG. 7) of the multi-cloud infrastructure. Upon receiving the request (e.g., request to create an Exa-database as shown in FIG. 10), the proxy module may validate the user based on a token generated by the second cloud infrastructure as described previously. Upon successful validation, in some implementations, the proxy module passes the information obtained in the request to a corresponding adaptor(e.g., corresponding to the requested service) included in the pool of adaptors included in the multi-cloud infrastructure. For example, the

request 1000 of FIG. 10 may be directed by the proxy module to a database adaptor included in the multi-cloud infrastructure. In turn, the database adaptor upon receiving the request may make a call to the cloud-link adaptor of the multi-cloud infrastructure in order to obtain mapping information related to mapping between the account of the user in the second cloud infrastructure to the tenancy associated with the user in the first cloud infrastructure.

**[0191]** It is noted that if the mapping information does not exist, the cloud-link adaptor rejects the user's request 1000. In contrast, if the mapping information exists, the cloud-link adaptor provides the mapping information (as well as the resource-principal associated with the resource) to the database adaptor, whereafter, the database adaptor issues a request to the downstream service e.g., database-as-a-service (DBaaS), to execute the request.

**[0192]** FIG. 11 illustrates an exemplary flowchart depicting steps performed by a multi-cloud infrastructure, according to certain embodiments. The processing depicted in FIG. 11 may be implemented in software (e.g., code, instructions, program) executed by one or more processing units (e.g., processors, cores) of the respective systems, hardware, or combinations thereof. The software may be stored on a non-transitory storage medium (e.g., on a memory device). The method presented in FIG. 11 and described below is intended to be illustrative and non-limiting. Although FIG. 11 depicts the various processing steps occurring in a particular sequence or order, this is not intended to be limiting. In certain alternative embodiments, the steps may be performed in some different order or some steps may also be performed in parallel.

**[0193]** The process commences in step 1105, where a first cloud services provider provides, using a first cloud infrastructure, a set of more of more cloud services. In step 1110, a multi-cloud infrastructure is provided within the first cloud infrastructure. The multi-cloud infrastructure enables a user associated with an account with a second cloud services provider to use, from a second cloud infrastructure, a first service from the set of one or more cloud services. In other words, the multi-cloud infrastructure forms a gateway for users of other cloud services providers to utilize the one or more cloud services provided by the first cloud infrastructure from their native cloud environments. It is appreciated that the first service may correspond to creating and utilizing a database resource that is deployed in the first cloud infrastructure (e.g., in a tenancy associated with the user and created in the first cloud infrastructure).

**[0194]** The process then moves to step 1115, where the multi-cloud infrastructure creates a link between the account (of the user) with the second cloud services provider and a tenancy

created in the first cloud infrastructure for enabling the user to use the first service. In some implementations, the multi-cloud infrastructure generates a data object that stores metadata information corresponding to the link. The metadata information may include a mapping of a first identifier associated with the tenancy created in the first cloud infrastructure and a second identifier associated with the account (of the user) with the second cloud services provider.

**Establishing Tenancy In A Cloud Environment**

**[0195]** As described previously with reference to FIG. 7, the cloud-link adaptor 722D included in the multi-cloud infrastructure 720B is responsible for handling lifecycle operations of resources provided by the first cloud infrastructure. The cloud-link adaptor 722D is configured to create a mapping between an active directory tenant of the second cloud infrastructure (and its associated subscriptions) and a corresponding tenancy/account of the user in the first cloud infrastructure. In other words, the cloud-link adaptor generates a mapping of a first identifier associated with the tenancy of the user in the first cloud infrastructure to a second identifier associated with the account of the user in the second cloud infrastructure. It is noted that linking the tenancy of the user in the first cloud infrastructure to the account of the user in the second cloud infrastructure enables the user to utilize a service provided by the first cloud infrastructure using identity information of the user associated with the second cloud infrastructure.

**[0196]** By some embodiments, the cloud-link adaptor 722D creates a data object (referred to herein as a cloud-link resource object or a link-resource object) that that includes information linking the tenancy of the user in the first cloud infrastructure to the account of the user in the second cloud infrastructure. FIG. 12 illustrates an exemplary relationship diagram of a link-resource object, according to certain embodiments. As shown in the relationship diagram 1200, a cloud link resource object 1210 links a tenancy in a first cloud environment 1215 (e.g., customer tenancy 726 of FIG. 7) to an active directory tenant in a second cloud environment 1225. The active directory tenant in the second cloud environment is associated with a subscription/account of a user in the second cloud environment (e.g., customer subscription 715 of FIG. 7).

**[0197]** It is appreciated that the user subscription in the second cloud environment 1220 has a 1:1 relationship with the tenancy of the user in the first cloud environment 1215. The cloud-link resource object 1210 includes a mapping of a first identifier associated with the tenancy of the user in the first cloud infrastructure (e.g., tenancy name in the first cloud environment) to a second identifier associated with the account of the user in the second cloud infrastructure

(e.g., tenancy ID and associated subscription in the second cloud environment). Furthermore, the cloud-link resource object 1210 also stores location information 1230 pertaining to region/location of the first cloud environment and the second cloud environment, where the linkage between the user accounts in the two cloud environments is enabled.

**[0198]** FIG. 13 depicts a flow diagram illustrating an example process of linking two user accounts in different cloud environments, according to certain embodiments. For example, the processing depicted in FIG. 13 may be applied to link a user's Azure account (i.e., account in the second cloud environment ) with the user's Oracle account (i.e., account in the first cloud environment). When a customer creates a tenancy 1372 in the second cloud environment 1370, an active directory 1373 is associated with the tenancy 1372. Note that the customer may create the tenancy in the second cloud environment via a second cloud portal (e.g., portal 711 of FIG. 7). The active directory 1373 is used for managing identities associated with the customer's tenancy 1372 including managing identities associated with users associated with the customer's tenancy 1372. In FIG. 13, the active directory 1373 is shown for a particular customer's tenancy 1373.

**[0199]** In some implementations, the active directory 1373 stores the following pieces of information that are used for identity management within the second cloud environment:

(1) Applications Information – comprises information identifying one or more applications that have been registered with the tenancy 1372 in the second cloud environment, and which can be used by users associated with the tenancy 1372 . For example, in FIG. 13, the active directory 1373 stores information identifying a multi-cloud application 1309 that has been registered for the customer's tenancy 1372. For each application, one or more roles for that application may be specified. For example, in FIG. 14, the roles associated with the multi-cloud application 1309 may include a first role (Role 1) that corresponds to an administrator role, and a second role (Role 2) that corresponds to a reader. It is noted that a role can be associated with one or more users. As shown in FIG. 13, users 1308 includes a first user (user A) who is associated with Role 1, and a second user (user B) who is associated with Role 2. The access privileges for Role 2 may be different (e.g., may be fewer) than those for Role 1. For example, while Role 1 may permit various operations (e.g., CRUD operations) on a resource (e.g., database), Role 2 may only permit a read operation with respect to the resource.

(2) User Information – comprises information identifying a set of one or more users associated with the tenancy 1372 in the second cloud environment. For example, as shown in FIG. 13, customer's tenancy 2072 includes users 1308 comprising two users i.e., user A and

user B. As described below, user A may correspond to an administrative user who may request a new account to be created (in the first cloud environment) for the customer and to link the new account to the customer's tenancy 2072. Additionally, user A may also request an existing account in the first cloud environment to be linked to the account in the second cloud environment.

(3) Groups Information (Groups 1306) – One or more groups 1306 can be configured for the tenancy 1372 and information identifying these groups 1306 is included in the active directory 1373 for the tenancy 1372. For example, as shown in FIG. 13, the groups information 1306 includes an administrative group comprising user A. It is noted that a group can include one or more users associated with the tenancy 1372, and a particular user may belong to different groups.

(4) Roles Information (AD Roles 1307) – includes information identifying one or more roles that have been configured for the tenancy 1372. AD Roles 1307 are considered global roles. Each AD role identifies tasks/operations/activities that are permitted under that role. In the example in FIG. 13, an "admin" role has been configured for the tenancy 1372. A role can be associated with one or more users. When a role is associated with a user, the role identifies the permitted tasks/operations/activities for that user. For example, in FIG. 13, user A is shown as being associated with the "admin" role in AD Roles 1307. If an "admin" role is associated with a user identified in the set of users 2008, then that user is considered an "admin" and is allowed to perform tasks/operations/activities permitted under the admin role. A user in the set of users 2008 can be associated with one or multiple roles from the AD Roles 1307. A role from the AD Roles 1307 can be associated with a group. When a particular role is associated with a group, the one or more users in that group are permitted to perform tasks/operations/activities that are permitted for that role.

**[0200]** In what follows, there is described with reference to FIG. 13, processing that may be performed for linking the tenancy of the user in the first cloud environment 1320 to the account of the user in the second cloud environment 1370. The processing depicted in FIG. 13 and described below is provided as an example and is not meant to be limiting. The processing may vary in certain implementations.

**[0201]** Processing may be initiated, at step 1, when a user (e.g., user A in FIG. 13) makes a call (i.e., a request) to the first cloud environment from the tenancy 1372 (in the second cloud environment) requesting the creation of a new tenancy in the first cloud environment, and further requesting a linking of the newly created tenancy in the first cloud environment to the

account 1372 associated with the user. It is appreciated that user A may initiate such a call by using the multi-cloud console (e.g., console 721 depicted in FIG. 7). The request may be processed by an account services component 1321 within the first cloud environment. In certain implementations, user A may initiate the call by calling an API provided by account services 1321. As part of the call, user A can provide identity information of the user corresponding to the second cloud environment. Such identity information may be in the form of a token (e.g., a bearer token that is generated by the second cloud environment). The token can provide relevant information about the requesting user, such as the user's identity related information in the active directory 1373 of the second cloud environment. The identity related information may include information about roles and group affiliations of the requesting user within active directory 1373.

**[0202]** Responsive to receiving the request, account services 1321 may proceed to validate the request. In some implementations, account services 1321 may transmit the request to a multi-cloud infrastructure 1350 included in the first cloud environment 1320. Specifically, as described previously with reference to FIG. 7, an authority module 1336 that is included in the multi-cloud infrastructure 1350 may validate the request. The token received in the request may be used to validate the user. Accordingly, the identity associated with the requesting user (e.g., user A) is used to validate the user. As part of the validation, processing may be performed to verify that user A is really a member of, or a user associated with the customer's account 1372. Further, validation processing may include verifying one or more roles associated with the user to determine whether the user has sufficient privileges to issue a request pertaining to creation of a new tenancy in the first cloud environment.

**[0203]** Once the verification of the user is successful, at step 3, account services 1321 proceeds to create a new tenancy/account 1322 in the first cloud environment 1320. The tenancy 1322 may have certain default configurations. For example, a domain 1323 is associated with tenancy 1322. Like the active directory 1373 on the second cloud environment side, the domain 1323 associated with the tenancy in the first cloud environment facilitates the management of identities. A domain 1323 may include users, groups, etc., associated with the tenancy 1322. In certain implementations, by default, when account/tenancy 1322 is created, the domain 1323 may include an "admin" group 1324, which includes a user, such as "User 1" as shown in FIG. 13. User 1's identity may be created for the requesting user (i.e., user A). However, as discussed below, one of the reasons for linking the tenancy of the user in the first cloud infrastructure to the account of the user in the second cloud infrastructure is to enable the

user to use resources in the first cloud environment using the identity information of the user in the second cloud environment, such that the user does not have to remember or even know about the "User 1" identity created on the first cloud environment side. Accordingly, user A can access the multi-cloud console (using identity information related to the second cloud environment) and trigger a request to create a tenancy in the first cloud environment.

**[0204]** Once the tenancy 1322 has been created in the first cloud environment, processing is then performed to link the newly created tenancy 1322 with the account 1372 associated with the user. In the example implementation depicted in FIG. 13, at step 4, account services 1321 transmits a request to the cloud-link adaptor 1330 included in the multi-cloud infrastructure 1350 to perform the linking process. Such a request may include in its payload, information regarding the two accounts to be linked, the token generated by the second cloud environment, etc. As described below, as part of the linkage related processing, processing may be performed to enable the user to, access endpoints/APIs provided by first cloud environment (e.g., to access, use, and manage resources in the tenancy associated with the user in the first cloud environment), by using login/identity information of the user in the second cloud environment.

**[0205]** Steps 5 and 6, that are performed by the cloud-link adaptor 1330 pertain to the linking process. At step 5, the cloud-link adaptor 1330 creates a data object (i.e., a link-resource object/ or a cloud-link resource object) 1331 that represents the link between the two accounts. Specifically, as shown in FIG. 12, the link-resource object includes information linking the tenancy of the user in the first cloud infrastructure to the account of the user in the second cloud infrastructure. The link-resource object enables the user to utilize one or more services provided by the first cloud environment. In some implementations, linking the tenancy of the user in the first cloud environment to the account of the user in the second cloud environment includes storing, in the link-resource object, a mapping of a first identifier associated with the tenancy of the user in the first cloud environment to a second identifier associated with the account of the user in the second cloud environment. In certain implementations, the cloud-link resource object 1331 created in step 5 is stored in a compartment (e.g., root compartment) associated with the tenancy 1322 included in the first cloud environment.

**[0206]** At step 6, cloud-link adaptor 1330 creates a resource-principal (also referred to herein as a cloud-link resource principal) 1332 that is associated with the cloud-link resource object 1331. When a user transmits (from the second cloud environment), a request to the multi-cloud infrastructure (e.g., via the multi-cloud console) to create/manage a resource in the tenancy of the user in the first cloud environment, the multi-cloud infrastructure uses the token included

in the request to validate the user. As the token is generated by the second cloud environment, it is unrecognizable by services provided by the first cloud environment. Thus, the cloud-link adaptor 1330 creates the resource-principal (that functions as a deputy or a valid identity information corresponding to the resource, and which is generated by the multi-cloud infrastructure included in the first cloud environment), which is utilized by a downstream service of the first cloud environment to execute the request.

**[0207]** In order for the resource principal to enable the downstream service to be executed in a seamless manner, permissions are associated with the resource principal based upon the user's credentials/token in the second cloud environment. Consent for setting up the resource principal is provided by the user when the user issues a request to the multi-cloud infrastructure via the multi-cloud console. Furthermore, in some implementations, the resource-principal enables the first cloud environment 1320 to have access to the account of the user in the second cloud environment. For instance, the resource-principal may be used to set up federation between the two user accounts in the different cloud environments.

**[0208]** Further, in step 7, as shown in FIG. 13, a federation may be established between the user's accounts in the first cloud environment and the second cloud environment. As a result of the federation, users and groups associated with the active directory 1373 in the second cloud environment are synced to the tenancy 1322 included in the first cloud environment. For instance, in one implementation, a new group may be created in the active directory 1373 in the second cloud environment. The user is then added to this newly created group. Further, information pertaining to the newly created group in the second cloud environment is copied or synchronized over to the tenancy of the customer in the first cloud environment. As a result, a new group (e.g., administrative group) is created in a domain of the customer's tenancy in the first cloud environment, with the user included in this group (in the first cloud environment). Further, in some implementations, as part of the federation process, the cloud-link adaptor 1330 may create a policy and associate the policy with the newly created group. In certain implementations, the policy permits full access (e.g., admin-level access) with the first cloud environment. This causes the group created in the first cloud environment (along with the users included in the group) to have full access within customer's tenancy in the first cloud environment. As a result, the user can now perform administrative tasks within the tenancy in the first cloud environment.

**[0209]** Referring to FIG. 14, there is depicted a flow diagram illustrating an example process of using a cloud-link resource object in deploying resources, according to certain embodiments.

The processing depicted in FIG. 14 and described below is provided as an example and is not meant to be limiting. The processing may vary in certain implementations. Upon the customer's account in the first cloud environment (e.g., tenancy 1322) being successfully linked to the customer's tenancy/account in the second cloud environment (e.g., tenancy 1372), a user may initiate a process to create, use, and manage resources provided by the first cloud environment. In some implementations, the user may initiate a request to the multi-cloud console to create a resource in the tenancy associated with the user in the first cloud environment. Such a request is routed by the multi-cloud console to the multi-cloud infrastructure included in the first cloud environment. Alternatively, in some implementations, the user may call, or access one or more endpoints provided by multi-cloud infrastructure. For example, user A may call APIs provided by multi-cloud infrastructure using a token that is generated in the second cloud environment. User A can issue such an API call to the multi-cloud infrastructure endpoint using publicly accessible REST APIs.

**[0210]** As shown in FIG. 14, user A initiates a call requesting a resource to be deployed in the tenancy associated with the user in the first cloud environment (Step 1). Such a request may be routed from the multi-cloud console to the accounts service module 1321 included in the first cloud environment. As part of the request, the user A may provide a token (generated by the second cloud environment) that identifies user A and/or any other relevant account, group, or customer information. In general, when a user calls an endpoint provided by first cloud environment from the customer's account 1372 (in the second cloud environment) to perform a certain operation with respect to a resource, on the first cloud environment side, before the requested operation is performed, the first cloud environment performs validation to: (a) check whether the user requesting the operation is a proper user associated with the customer's account 1372, and (b) check that the user is associated with a role that permits the user to perform the requested operation.

**[0211]** At step 2, after user A issues the request to create a new resource (e.g., a database resource), accounts services 1321 forwards the request to the multi-cloud infrastructure 1350 to validate the request by checking whether user A is associated with a particular role on the second cloud environment side, where the role permits the user to create the requested resource. The processing of validation performed in step 2 of FIG. 14 is similar to the validation processing performed in step 2 of FIG. 13. For instance, authority module of the multi-cloud infrastructure 1350 checks if the user is associated with an administrative role created under the multi-cloud application 1309 in the customer's tenancy 1372, and further checks that the

administrative role allows creation of the resource. If multi-cloud infrastructure determines that user A is associated with the administrative role in the active directory 1373 and that the role permits a resource (e.g., database) to be created, then the user is considered to be successfully validated. It is noted that in the above validation process, in order for the multi-cloud infrastructure to obtain information pertaining to the roles associated with a given token, the authority module included in the multi-cloud infrastructure may communicate with the active directory in the second cloud environment to obtain the information.

**[0212]** Upon successful validation of the user, the proxy module included in the multi-cloud infrastructure may transmit the token (included in the request made via the multi-cloud console) to an adaptor (e.g., database adaptor corresponding to a service requested by the user). The database adaptor 1337 in turn queries the cloud-link adaptor 1330 (in step 4) to obtain mapping information of the user accounts in the two cloud environments. The cloud-link adaptor 1330 can then identify (based on the token), the appropriate tenancy 1322 that is associated with token e.g., the customer identity, user identity, etc. As a result, an API call that includes identity information (e.g., the token) of the user with respect to the second cloud environment and does not include specific identity information related to the first cloud environment, can be used to lookup a corresponding tenancy 1322 in the first cloud environment. Upon identifying a particular tenancy 1322 that is linked to the tenancy/account 1372 in the second cloud environment, the cloud-link adaptor 1330 can also retrieve from the tenancy 1322, the cloud-link resource principal that is associated with the cloud-link resource object. The cloud-link adaptor 1330 transmits the retrieved information back to the database adaptor 1337.

**[0213]** At step 5, the database adaptor 1337 transmits a request to the corresponding downstream service offered by the first cloud environment (e.g., DBaaS 1360). The request pertains to the creation of a resource (e.g., database) in the tenancy associated with the user in the first cloud environment. It is appreciated that the request includes the resource-principal (obtained by the database adaptor in step 4). The DBaaS service 1360 can then determine, based on the resource-principal (i.e., based on policies associated with the resource-principal), whether creation of a database within the first cloud environment is permitted for the user. It is notes that according to some embodiments, the same resource-principal that is stored in the customer's tenancy 1322 in the first cloud environment can be used for all users associated with that customer's account 1372.

Further, in step 6, upon the DBaaS service 1360 successfully authorizing the user, the DBaaS service 1360 proceeds to create the desired resource (e.g., database) in the tenancy associated

with the user in the first cloud environment. In some implementations, upon the resource being deployed in the customer tenancy in the first cloud environment, the multi-cloud infrastructure provides a web-link corresponding to the created resource, to the user in the second cloud environment. For instance, the web-link may be displayed in an application that the user executes (in the subscription/account) in the second cloud environment. In this manner, the user may directly monitor (in the second cloud environment), performance evaluation parameters related to the resource (e.g., database usage metrics) deployed in the first cloud environment.

**[0214]** Thus, embodiments of the present disclosure provide for a process of using a cloud-link resource object for connecting two cloud environments, such that a user of a first cloud can utilize services from a second cloud from within the first cloud's user-interface. The user can authenticate with the first cloud, and then access the second cloud's services without having to separately authenticate or provide identity information at the second cloud. Also, the user's access levels and permissions (e.g., admin-level vs. reader-level) at the first cloud can effectively be translated into access permissions downstream at the second cloud (e.g., the cloud-link resource principal). The cloud-link data object storing metadata linking the two cloud accounts (and detailing user access levels) can be used to automatically allow the user to access the second cloud and permitted second cloud resources without requiring the user to provide further authentication or identity information. Additionally, embodiments provide a framework for establishing a tenancy or account for the user at the second cloud without requiring the user to directly establish the account or setup a second set of credentials. In other words, the second account (with the second cloud) can be automatically created on behalf of the user without requiring the user's participation.

**Identity –Token Exchange**

**[0215]** In the above-described frameworks of FIGS. 13 and 14, the first cloud environment (i.e., a service providing cloud environment) integrates with an external cloud provider (e.g., second cloud environment or also referred to as a service caller's cloud environment) for authentication and authorization purposes. Once the call is authenticated and authorized, based on a token issued by the service caller's cloud environment (e.g., second cloud environment), the caller's context (i.e., user's context) is not carried forward for any downstream service calls that are made internally in the first cloud environment. Rather, as described above with reference to FIG. 14, all downstream service calls are made using a resource-principal that has valid access to a target service, on behalf of the caller.

**[0216]** According to some embodiments, it may be beneficial to preserve an identity context, mapping the user in the service caller's cloud environment to an equivalent principal (e.g., user principal) in the service providing cloud environment for the purposes of audit. Accordingly, by some embodiments, there is described a mechanism that provisions for exchanging a source cloud's token (e.g., issued in the service caller's cloud environment) to another token that is valid in the in the service providing cloud environment, while preserving the user's context for downstream calls.

**[0217]** According to some embodiments, the framework for exchanging a token generated by the service caller's cloud environment (e.g., token issued by second cloud environment of FIG. 13) to another token (i.e., a token usable by service providing cloud environment) relies on the service providing cloud environment's identity-as-a-service (IDaaS) support in achieving user identity propagation within its cloud environment. Identity propagation may be achieved via a token exchange i.e., standard based tokens (e.g., JWT or SAML assertions) generated by the service caller's cloud environment can be exchanged for internal principals (of the service providing cloud environment) like user principal session token (UPST), which in turn can be used for downstream service calls made in the service providing cloud environment.

**[0218]** In order to allow for such a token exchange, trust is to be established between an external identity provider (e.g., service caller's cloud environment) and the service providing cloud environment. In some implementations, trust is established via a configuration (referred to herein as cloud-identity-propagation configuration), that is introduced (e.g., by an administrator) in an identity and access management system of the service providing cloud environment. According to some embodiments, the configuration includes a plurality of attributes including: a provider name attribute, a provider type attribute, a provider issuer attribute, a subject of token attribute (corresponding to an actor who is authenticated, wherein the actor may be a user or an application), a subject mapping attribute name (corresponding to a lookup of the subject in the identity and access management system of the service providing cloud environment), a claims attribute including a maximum session duration etc. Further, in the remainder of the present disclosure (and for sake of uniformity), the service providing cloud environment is referred to as the first cloud environment and the service caller's cloud environment is referred to as the second cloud environment or an external identity provider system.

**[0219]** FIG. 15 depicts a flow diagram illustrating steps performed in a token exchange process, according to certain embodiments. FIG. 15 depicts interaction between a user 1510, a multi-cloud console 1515, an identity and access management (IAM) system of the first cloud services provider 1520, a multi-cloud infrastructure 1525 included in the first cloud environment, a second cloud provider 1530 (e.g., a third-party cloud services provider), and services offered by the first cloud environment 1535. The steps of flow depicted in FIG. 15 can be described as follows:

**[0220]** In step 1, user 1510 accesses the multi-cloud console 1515.

**[0221]** In step 2, the multi-cloud console 1515 redirects the user 1510 to a login API of the second cloud services provider 1530.

**[0222]** In step 3, the user 1510 performs a login operation with the second cloud services provider 1530.

**[0223]** In step 4, upon successful user login, the second cloud services provider 1530 returns a first token (generated by the second cloud services provider) to the multi-cloud console 1515.

**[0224]** In step 5, the multi-cloud console 1515 calls an API of the multi-cloud infrastructure 1525 using the first token obtained in step 4.

**[0225]** In step 6, the multi-cloud infrastructure 1525 validates the first token. By some embodiments, such a validation can be performed by an authority module (e.g., authority module 722A of FIG. 7) included in a control plane of the multi-cloud infrastructure.

**[0226]** In step 7, the multi-cloud infrastructure 1525 may obtain a public key from the second cloud services provider and may optionally issue a call to the second cloud services provider in order to obtain cloud provider specific business functionality.

**[0227]** In step 8, the multi-cloud infrastructure 1525 performs a token exchange request with the IAM system 1520 (of the first cloud services provider) using the first token in order to obtain a user principal service token (UPST) i.e., a second token that is usable by a service provided by the first cloud. Note that the first token issued by the second cloud services provider of step 4 is not usable by services provided in the first cloud. Further, token exchange request may be a signed request where the payload includes the first token obtained by the multi-cloud infrastructure in step 5.

**[0228]** In step 9, IAM system of the first cloud services provider 1520 validates the first token (i.e., token generated by the second cloud services provider). It is noted that validation may be performed by analyzing standard claims such as a signature of the token, expiration feature of the token, etc. It is noted that the token exchange request received by the IAM system

1520 may be a signed request. Accordingly, the IAM system 1520 validates the token exchange request based on a public key obtained from the second cloud infrastructure (i.e., second CSP) and a configuration that is previously setup between the first CSP and the second CSP. The configuration includes information mapping an identity of the user in the second cloud infrastructure to a corresponding user identity in the first cloud infrastructure.

**[0229]** In some implementations, the IAM system 1520 performs the following sub-processes: (a) IAM analyzes the configuration based on an issuer and/or entity ID i.e., IAM performs processing to verify if a known entity is requesting a token exchange. If no such configuration exists, then the token exchange request is rejected, (b) IAM validates the user by performing a lookup using 'subject' claim value and based on subject mapping attribute. Specifically, IAM system 1520 analyzes a subject attribute and/or a subject mapping attribute included in the configuration to determine existence of the user in a domain of the IAM system. Here two cases are possible- user exists in the IAM domain or the user does not exist in the IAM domain. In the first case, where the user exists in the IAM domain, the IAM system successfully validates the user. In the second case, where the user does not exist in the IAM domain, a just-in-time user provisioning is initiated by the IAM system, where the IAM system creates an account for the user in the first cloud.

**[0230]** In step 10, IAM system 1520 generates the user principal session token (UPST) i.e., the second token, and returns it to the multi-cloud infrastructure 1525. According to some embodiments, the second token's lifetime is determined based on the request's specified time or based on configuration value, or a default value, whichever is the minimum. In step 11, multi-cloud infrastructure 1525 makes a call to a downstream service of the first cloud infrastructure using the second token obtained in step 10. In step 12, multi-cloud infrastructure 1525 returns a response message (with respect to the requested service) to the multi-cloud console 1515.

**[0231]** FIG. 16 depicts a schematic illustrating an exemplary token exchange process, in accordance with some embodiments. Specifically, FIG. 16 illustrates a flow diagram depicting a process of performing a token exchange i.e., for a user associated with an account in a second cloud infrastructure and requesting use of a service provided by a first cloud infrastructure, the schematic of FIG. 16 illustrates how a first token (generated by a second cloud services provider providing the second cloud infrastructure) is exchanged for a second token that is generated by the first cloud infrastructure and usable by services provided by the first cloud infrastructure. Note that the first token generates by the second cloud infrastructure is not

usable by service provided by the first cloud infrastructure. Thus, in order for the user of the second cloud infrastructure to use one or more services provided by the first cloud infrastructure, a token exchange is required.

**[0232]** In some implementations, the process of exchanging tokens is based on establishing trust between the first cloud infrastructure and the second cloud infrastructure. As shown in FIG. 16, trust may be set up between the first cloud and the second cloud based on a configuration (e.g., certificate exchange and user mapping) performed previously by an administrator. Moreover, the accounts of the user in the first cloud infrastructure and the second cloud infrastructure may be federated and synched by the administrator. In order to establish trust, 'token-exchange' applications may be defined on both identity providers (i.e., the first cloud infrastructure and the second cloud infrastructure) to exchange certificates, configure claims that are sent in the token and set up mapping rules for matching an incoming user to a user in the destination cloud. An administrator may create an application in the identity providing system of the first cloud infrastructure to provide the issuer certificate used by the identity provider of the second cloud infrastructure in order to sign tokens and set mapping rules to convert claims in a first token (e.g., generated by the second cloud infrastructure) to lookup a user in the first cloud infrastructure.

**[0233]** As shown in FIG. 16, a user 1605 of the second cloud environment may perform a login operation in the multi-cloud console 1610 based on a credentials of the user associated with the second cloud infrastructure. The multi-cloud console 1610 makes a call to a service endpoint 1625 of the multi-cloud service. Such a call may include a first token that is generated by the second cloud infrastructure. The service endpoint 1625 after authenticating and authorizing (based on the first token included the call) the user with the second cloud environment, makes a call to an IAM token exchange endpoint 1630 to obtain a second token that is understandable and usable in the first cloud infrastructure.

**[0234]** The second token identifies a user principal in a target tenancy (of the first cloud) that is mapped to the incoming user, as long as the incoming user is synched and the identity provider of the first cloud allows tokens (e.g., on-behalf-of (OBO) tokens) to be issued for its users. According to some embodiments, the OBO token will be signed by the IAM service in the first cloud, so it is usable by various endpoints in the first cloud to access resources based on access policies in the tenancy. The second token can be signed by the IAM system of the first cloud and contain a public key of an ephemeral key pair that is used by the calling service to make signed requests to downstream resources of the first cloud infrastructure. In some

implementations, the second token also contains the key of the issuer and a subject of the first token so that it can identify the mapped caller in the second cloud for auditing purposes.

**[0235]** In this manner, the identity of the user (in the second cloud infrastructure) is mapped to a target user in the first cloud infrastructure and propagated downstream to services offered by the first cloud infrastructure. The identity of the user is authorized downstream based on access granted to the target user in the first cloud infrastructure and is used for auditing requests made to downstream resources. These audits can be used to authoritatively identify the original caller (i.e., the user in the second cloud infrastructure) to mitigate repudiation or thwart any confused deputy threats that are inherent to the existing flows where identities are not propagated.

**[0236]** Turning to FIG. 17, there is depicted a flowchart illustrating a process performed by a multi-cloud infrastructure, according to certain embodiments. The processing depicted in FIG. 17 may be implemented in software (e.g., code, instructions, program) executed by one or more processing units (e.g., processors, cores) of the respective systems, hardware, or combinations thereof. The software may be stored on a non-transitory storage medium (e.g., on a memory device). The method presented in FIG. 17 and described below is intended to be illustrative and non-limiting. Although FIG. 17 depicts the various processing steps occurring in a particular sequence or order, this is not intended to be limiting. In certain alternative embodiments, the steps may be performed in some different order or some steps may also be performed in parallel.

**[0237]** The process commences in step 1705, where a multi-cloud infrastructure included in a first cloud infrastructure (provided by a first cloud services provider) receives a first request from a user associated with an account in a second cloud infrastructure (provided by a second cloud services provider). Note that the first cloud infrastructure is different than the second cloud infrastructure. The first request corresponds to a request for using a service provided by the first cloud infrastructure and includes a first token that is generated by the second cloud infrastructure.

**[0238]** In step 1710, the multi-cloud infrastructure validates the first token issued by the second cloud infrastructure. It is noted that the first token is unusable by any service provided by the first cloud infrastructure. Accordingly, in order for a user of the second cloud infrastructure to use any service provided by the first cloud infrastructure, a token exchange is required, where the first token is exchanged for a second token that is generated by the first cloud infrastructure and usable by services provided by the first cloud infrastructure.

[0239] In step 1715, the multi-cloud infrastructure obtains the second token from an identity management system of the first cloud infrastructure based on a configuration previously setup (e.g., by an administrator) between the first cloud infrastructure and the second cloud infrastructure. Further, in step 1720, the multi-cloud infrastructure transmits the second token (obtained in step 1715) to the service provided by the first cloud infrastructure, wherein because the second token is generated (and signed) by the identity management system of the first cloud infrastructure, the second token is understandable (and usable) by the service provided by the first cloud infrastructure and thus enables the user of the second cloud infrastructure to utilize the service.

**Examples of Cloud Infrastructure**

[0240] As noted above, infrastructure as a service (IaaS) is one particular type of cloud computing. IaaS can be configured to provide virtualized computing resources over a public network (e.g., the Internet). In an IaaS model, a cloud computing provider can host the infrastructure components (e.g., servers, storage devices, network nodes (e.g., hardware), deployment software, platform virtualization (e.g., a hypervisor layer), or the like). In some cases, an IaaS provider may also supply a variety of services to accompany those infrastructure components (e.g., billing, monitoring, logging, security, load balancing and clustering, etc.). Thus, as these services may be policy-driven, IaaS users may be able to implement policies to drive load balancing to maintain application availability and performance.

[0241] In some instances, IaaS customers may access resources and services through a wide area network (WAN), such as the Internet, and can use the cloud provider's services to install the remaining elements of an application stack. For example, the user can log in to the IaaS platform to create virtual machines (VMs), install operating systems (OSs) on each VM, deploy middleware such as databases, create storage buckets for workloads and backups, and even install enterprise software into that VM. Customers can then use the provider's services to perform various functions, including balancing network traffic, troubleshooting application issues, monitoring performance, managing disaster recovery, etc.

[0242] In most cases, a cloud computing model will require the participation of a cloud provider. The cloud provider may, but need not be, a third-party service that specializes in providing (e.g., offering, renting, selling) IaaS. An entity might also opt to deploy a private cloud, becoming its own provider of infrastructure services.

[0243] In some examples, IaaS deployment is the process of putting a new application, or a new version of an application, onto a prepared application server or the like. It may also include

the process of preparing the server (e.g., installing libraries, daemons, etc.). This is often managed by the cloud provider, below the hypervisor layer (e.g., the servers, storage, network hardware, and virtualization). Thus, the customer may be responsible for handling (OS), middleware, and/or application deployment (e.g., on self-service virtual machines (e.g., that can be spun up on demand) or the like.

**[0244]** In some examples, IaaS provisioning may refer to acquiring computers or virtual hosts for use, and even installing needed libraries or services on them. In most cases, deployment does not include provisioning, and the provisioning may need to be performed first.

**[0245]** In some cases, there are two different problems for IaaS provisioning. First, there is the initial challenge of provisioning the initial set of infrastructure before anything is running. Second, there is the challenge of evolving the existing infrastructure (e.g., adding new services, changing services, removing services, etc.) once everything has been provisioned. In some cases, these two challenges may be addressed by enabling the configuration of the infrastructure to be defined declaratively. In other words, the infrastructure (e.g., what components are needed and how they interact) can be defined by one or more configuration files. Thus, the overall topology of the infrastructure (e.g., what resources depend on which, and how they each work together) can be described declaratively. In some instances, once the topology is defined, a workflow can be generated that creates and/or manages the different components described in the configuration files.

**[0246]** In some examples, an infrastructure may have many interconnected elements. For example, there may be one or more virtual private clouds (VPCs) (e.g., a potentially on-demand pool of configurable and/or shared computing resources), also known as a core network. In some examples, there may also be one or more security group rules provisioned to define how the security of the network will be set up and one or more virtual machines (VMs). Other infrastructure elements may also be provisioned, such as a load balancer, a database, or the like. As more and more infrastructure elements are desired and/or added, the infrastructure may incrementally evolve.

**[0247]** In some instances, continuous deployment techniques may be employed to enable deployment of infrastructure code across various virtual computing environments. Additionally, the described techniques can enable infrastructure management within these environments. In some examples, service teams can write code that is desired to be deployed to one or more, but often many, different production environments (e.g., across various

different geographic locations, sometimes spanning the entire world). However, in some examples, the infrastructure on which the code will be deployed must first be set up. In some instances, the provisioning can be done manually, a provisioning tool may be utilized to provision the resources, and/or deployment tools may be utilized to deploy the code once the infrastructure is provisioned.

**[0248]** FIG. 18 is a block diagram 1800 illustrating an example pattern of an IaaS architecture, according to at least one embodiment. Service operators 1802 can be communicatively coupled to a secure host tenancy 1804 that can include a virtual cloud network (VCN) 1806 and a secure host subnet 1808. In some examples, the service operators 1802 may be using one or more client computing devices, which may be portable handheld devices (e.g., an iPhone®, cellular telephone, an iPad®, computing tablet, a personal digital assistant (PDA)) or wearable devices (e.g., a Google Glass® head mounted display), running software such as Microsoft Windows Mobile®, and/or a variety of mobile operating systems such as iOS, Windows Phone, Android, BlackBerry 8, Palm OS, and the like, and being Internet, e-mail, short message service (SMS), Blackberry®, or other communication protocol enabled. Alternatively, the client computing devices can be general purpose personal computers including, by way of example, personal computers and/or laptop computers running various versions of Microsoft Windows®, Apple Macintosh®, and/or Linux operating systems. The client computing devices can be workstation computers running any of a variety of commercially available UNIX® or UNIX-like operating systems, including without limitation the variety of GNU/Linux operating systems, such as for example, Google Chrome OS. Alternatively, or in addition, client computing devices may be any other electronic device, such as a thin-client computer, an Internet-enabled gaming system (e.g., a Microsoft Xbox gaming console with or without a Kinect® gesture input device), and/or a personal messaging device, capable of communicating over a network that can access the VCN 1806 and/or the Internet.

**[0249]** The VCN 1806 can include a local peering gateway (LPG) 1810 that can be communicatively coupled to a secure shell (SSH) VCN 1812 via an LPG 1810 contained in the SSH VCN 1812. The SSH VCN 1812 can include an SSH subnet 1814, and the SSH VCN 1812 can be communicatively coupled to a control plane VCN 1816 via the LPG 1810 contained in the control plane VCN 1816. Also, the SSH VCN 1812 can be communicatively coupled to a data plane VCN 1818 via an LPG 1810. The control plane VCN 1816 and the data plane VCN 1818 can be contained in a service tenancy 1819 that can be owned and/or operated by the IaaS provider.

**[0250]** The control plane VCN 1816 can include a control plane demilitarized zone (DMZ) tier 1820 that acts as a perimeter network (e.g., portions of a corporate network between the corporate intranet and external networks). The DMZ-based servers may have restricted responsibilities and help keep security breaches contained. Additionally, the DMZ tier 1820 can include one or more load balancer (LB) subnet(s) 1822, a control plane app tier 1824 that can include app subnet(s) 1826, a control plane data tier 1828 that can include database (DB) subnet(s) 1830 (e.g., frontend DB subnet(s) and/or backend DB subnet(s)). The LB subnet(s) 1822 contained in the control plane DMZ tier 1820 can be communicatively coupled to the app subnet(s) 1826 contained in the control plane app tier 1824 and an Internet gateway 1834 that can be contained in the control plane VCN 1816, and the app subnet(s) 1826 can be communicatively coupled to the DB subnet(s) 1830 contained in the control plane data tier 1828 and a service gateway 1836 and a network address translation (NAT) gateway 1838. The control plane VCN 1816 can include the service gateway 1836 and the NAT gateway 1838.

**[0251]** The control plane VCN 1816 can include a data plane mirror app tier 1840 that can include app subnet(s) 1826. The app subnet(s) 1826 contained in the data plane mirror app tier 1840 can include a virtual network interface controller (VNIC) 1842 that can execute a compute instance 1844. The compute instance 1844 can communicatively couple the app subnet(s) 1826 of the data plane mirror app tier 1840 to app subnet(s) 1826 that can be contained in a data plane app tier 1846.

**[0252]** The data plane VCN 1818 can include the data plane app tier 1846, a data plane DMZ tier 1848, and a data plane data tier 1850. The data plane DMZ tier 1848 can include LB subnet(s) 1822 that can be communicatively coupled to the app subnet(s) 1826 of the data plane app tier 1846 and the Internet gateway 1834 of the data plane VCN 1818. The app subnet(s) 1826 can be communicatively coupled to the service gateway 1836 of the data plane VCN 1818 and the NAT gateway 1838 of the data plane VCN 1818. The data plane data tier 1850 can also include the DB subnet(s) 1830 that can be communicatively coupled to the app subnet(s) 1826 of the data plane app tier 1846.

**[0253]** The Internet gateway 1834 of the control plane VCN 1816 and of the data plane VCN 1818 can be communicatively coupled to a metadata management service 1852 that can be communicatively coupled to public Internet 1854. Public Internet 1854 can be communicatively coupled to the NAT gateway 1838 of the control plane VCN 1816 and of the data plane VCN 1818. The service gateway 1836 of the control plane VCN 1816 and of the data plane VCN 1818 can be communicatively couple to cloud services 1856.

**[0254]** In some examples, the service gateway 1836 of the control plane VCN 1816 or of the data plan VCN 1818 can make application programming interface (API) calls to cloud services 1856 without going through public Internet 1854. The API calls to cloud services 1856 from the service gateway 1836 can be one-way: the service gateway 1836 can make API calls to cloud services 1856, and cloud services 1856 can send requested data to the service gateway 1836. But, cloud services 1856 may not initiate API calls to the service gateway 1836.

**[0255]** In some examples, the secure host tenancy 1804 can be directly connected to the service tenancy 1819, which may be otherwise isolated. The secure host subnet 1808 can communicate with the SSH subnet 1814 through an LPG 1810 that may enable two-way communication over an otherwise isolated system. Connecting the secure host subnet 1808 to the SSH subnet 1814 may give the secure host subnet 1808 access to other entities within the service tenancy 1819.

**[0256]** The control plane VCN 1816 may allow users of the service tenancy 1819 to set up or otherwise provision desired resources. Desired resources provisioned in the control plane VCN 1816 may be deployed or otherwise used in the data plane VCN 1818. In some examples, the control plane VCN 1816 can be isolated from the data plane VCN 1818, and the data plane mirror app tier 1840 of the control plane VCN 1816 can communicate with the data plane app tier 1846 of the data plane VCN 1818 via VNICs 1842 that can be contained in the data plane mirror app tier 1840 and the data plane app tier 1846.

**[0257]** In some examples, users of the system, or customers, can make requests, for example create, read, update, or delete (CRUD) operations, through public Internet 1854 that can communicate the requests to the metadata management service 1852. The metadata management service 1852 can communicate the request to the control plane VCN 1816 through the Internet gateway 1834. The request can be received by the LB subnet(s) 1822 contained in the control plane DMZ tier 1820. The LB subnet(s) 1822 may determine that the request is valid, and in response to this determination, the LB subnet(s) 1822 can transmit the request to app subnet(s) 1826 contained in the control plane app tier 1824. If the request is validated and requires a call to public Internet 1854, the call to public Internet 1854 may be transmitted to the NAT gateway 1838 that can make the call to public Internet 1854. Memory that may be desired to be stored by the request can be stored in the DB subnet(s) 1830.

**[0258]** In some examples, the data plane mirror app tier 1840 can facilitate direct communication between the control plane VCN 1816 and the data plane VCN 1818. For example, changes, updates, or other suitable modifications to configuration may be desired to

be applied to the resources contained in the data plane VCN 1818. Via a VNIC 1842, the control plane VCN 1816 can directly communicate with, and can thereby execute the changes, updates, or other suitable modifications to configuration to, resources contained in the data plane VCN 1818.

**[0259]** In some embodiments, the control plane VCN 1816 and the data plane VCN 1818 can be contained in the service tenancy 1819. In this case, the user, or the customer, of the system may not own or operate either the control plane VCN 1816 or the data plane VCN 1818. Instead, the IaaS provider may own or operate the control plane VCN 1816 and the data plane VCN 1818, both of which may be contained in the service tenancy 1819. This embodiment can enable isolation of networks that may prevent users or customers from interacting with other users', or other customers', resources. Also, this embodiment may allow users or customers of the system to store databases privately without needing to rely on public Internet 1854, which may not have a desired level of security, for storage.

**[0260]** In other embodiments, the LB subnet(s) 1822 contained in the control plane VCN 1816 can be configured to receive a signal from the service gateway 1836. In this embodiment, the control plane VCN 1816 and the data plane VCN 1818 may be configured to be called by a customer of the IaaS provider without calling public Internet 1854. Customers of the IaaS provider may desire this embodiment since database(s) that the customers use may be controlled by the IaaS provider and may be stored on the service tenancy 1819, which may be isolated from public Internet 1854.

**[0261]** FIG. 19 is a block diagram 1900 illustrating another example pattern of an IaaS architecture, according to at least one embodiment. Service operators 1902 (e.g., service operators 1802 of FIG. 18) can be communicatively coupled to a secure host tenancy 1904 (e.g., the secure host tenancy 1804 of FIG. 18) that can include a virtual cloud network (VCN) 1906 (e.g., the VCN 1806 of FIG. 18) and a secure host subnet 1908 (e.g., the secure host subnet 1808 of FIG. 18). The VCN 1906 can include a local peering gateway (LPG) 1910 (e.g., the LPG 1810 of FIG. 18) that can be communicatively coupled to a secure shell (SSH) VCN 1912 (e.g., the SSH VCN 1812 of FIG. 18) via an LPG 1810 contained in the SSH VCN 1912. The SSH VCN 1912 can include an SSH subnet 1914 (e.g., the SSH subnet 1814 of FIG. 18), and the SSH VCN 1912 can be communicatively coupled to a control plane VCN 1916 (e.g., the control plane VCN 1816 of FIG. 18) via an LPG 1910 contained in the control plane VCN 1916. The control plane VCN 1916 can be contained in a service tenancy 1919 (e.g., the service tenancy 1819 of FIG. 18), and the data plane VCN 1918 (e.g., the data plane VCN 1818 of

FIG. 18) can be contained in a customer tenancy 1921 that may be owned or operated by users, or customers, of the system.

**[0262]** The control plane VCN 1916 can include a control plane DMZ tier 1920 (e.g. the control plane DMZ tier 1820 of FIG. 18) that can include LB subnet(s) 1922 (e.g. LB subnet(s) 1822 of FIG. 18), a control plane app tier 1924 (e.g. the control plane app tier 1824 of FIG. 18) that can include app subnet(s) 1926 (e.g. app subnet(s) 1826 of FIG. 18), a control plane data tier 1928 (e.g. the control plane data tier 1828 of FIG. 18) that can include database (DB) subnet(s) 1930 (e.g. similar to DB subnet(s) 1830 of FIG. 18). The LB subnet(s) 1922 contained in the control plane DMZ tier 1920 can be communicatively coupled to the app subnet(s) 1926 contained in the control plane app tier 1924 and an Internet gateway 1934 (e.g. the Internet gateway 1834 of FIG. 18) that can be contained in the control plane VCN 1916, and the app subnet(s) 1926 can be communicatively coupled to the DB subnet(s) 1930 contained in the control plane data tier 1928 and a service gateway 1936 (e.g. the service gateway of FIG. 18) and a network address translation (NAT) gateway 1938 (e.g. the NAT gateway 1838 of FIG. 18). The control plane VCN 1916 can include the service gateway 1936 and the NAT gateway 1938.

**[0263]** The control plane VCN 1916 can include a data plane mirror app tier 1940 (e.g., the data plane mirror app tier 1840 of FIG. 18) that can include app subnet(s) 1926. The app subnet(s) 1926 contained in the data plane mirror app tier 1940 can include a virtual network interface controller (VNIC) 1942 (e.g., the VNIC of 1842) that can execute a compute instance 1944 (e.g., similar to the compute instance 1844 of FIG. 18). The compute instance 1944 can facilitate communication between the app subnet(s) 1926 of the data plane mirror app tier 1940 and the app subnet(s) 1926 that can be contained in a data plane app tier 1946 (e.g., the data plane app tier 1846 of FIG. 18) via the VNIC 1942 contained in the data plane mirror app tier 1940 and the VNIC 1942 contained in the data plan app tier 1946.

**[0264]** The Internet gateway 1934 contained in the control plane VCN 1916 can be communicatively coupled to a metadata management service 1952 (e.g., the metadata management service 1852 of FIG. 18) that can be communicatively coupled to public Internet 1954 (e.g., public Internet 1854 of FIG. 18). Public Internet 1954 can be communicatively coupled to the NAT gateway 1938 contained in the control plane VCN 1916. The service gateway 1936 contained in the control plane VCN 1416 can be communicatively couple to cloud services 1956 (e.g., cloud services 1856 of FIG. 18).

**[0265]** In some examples, the data plane VCN 1918 can be contained in the customer tenancy 1921. In this case, the IaaS provider may provide the control plane VCN 1916 for each customer, and the IaaS provider may, for each customer, set up a unique compute instance 1944 that is contained in the service tenancy 1919. Each compute instance 1944 may allow communication between the control plane VCN 1916, contained in the service tenancy 1919, and the data plane VCN 1918 that is contained in the customer tenancy 1921. The compute instance 1944 may allow resources that are provisioned in the control plane VCN 1916 that is contained in the service tenancy 1919, to be deployed or otherwise used in the data plane VCN 1918 that is contained in the customer tenancy 1921.

**[0266]** In other examples, the customer of the IaaS provider may have databases that live in the customer tenancy 1921. In this example, the control plane VCN 1916 can include the data plane mirror app tier 1940 that can include app subnet(s) 1926. The data plane mirror app tier 1940 can reside in the data plane VCN 1918, but the data plane mirror app tier 1940 may not live in the data plane VCN 1918. That is, the data plane mirror app tier 1940 may have access to the customer tenancy 1921, but the data plane mirror app tier 1940 may not exist in the data plane VCN 1918 or be owned or operated by the customer of the IaaS provider. The data plane mirror app tier 1940 may be configured to make calls to the data plane VCN 1918 but may not be configured to make calls to any entity contained in the control plane VCN 1916. The customer may desire to deploy or otherwise use resources in the data plane VCN 1918 that are provisioned in the control plane VCN 1916, and the data plane mirror app tier 1940 can facilitate the desired deployment, or other usage of resources, of the customer.

**[0267]** In some embodiments, the customer of the IaaS provider can apply filters to the data plane VCN 1918. In this embodiment, the customer can determine what the data plane VCN 1918 can access, and the customer may restrict access to public Internet 1954 from the data plane VCN 1918. The IaaS provider may not be able to apply filters or otherwise control access of the data plane VCN 1918 to any outside networks or databases. Applying filters and controls by the customer onto the data plane VCN 1918, contained in the customer tenancy 1921, can help isolate the data plane VCN 1918 from other customers and from public Internet 1954.

**[0268]** In some embodiments, cloud services 1956 can be called by the service gateway 1936 to access services that may not exist on public Internet 1954, on the control plane VCN 1916, or on the data plane VCN 1918. The connection between cloud services 1956 and the control plane VCN 1916 or the data plane VCN 1918 may not be live or continuous. Cloud services 1956 may exist on a different network owned or operated by the IaaS provider. Cloud services

1956 may be configured to receive calls from the service gateway 1936 and may be configured to not receive calls from public Internet 1954. Some cloud services 1956 may be isolated from other cloud services 1956, and the control plane VCN 1916 may be isolated from cloud services 1956 that may not be in the same region as the control plane VCN 1916. For example, the control plane VCN 1916 may be located in "Region 1," and cloud service "Deployment 13," may be located in Region 1 and in "Region 2." If a call to Deployment 13 is made by the service gateway 1936 contained in the control plane VCN 1916 located in Region 1, the call may be transmitted to Deployment 13 in Region 1. In this example, the control plane VCN 1916, or Deployment 13 in Region 1, may not be communicatively coupled to, or otherwise in communication with, Deployment 13 in Region 2.

**[0269]** FIG. 20 is a block diagram 2000 illustrating another example pattern of an IaaS architecture, according to at least one embodiment. Service operators 2002 (e.g., service operators 1802 of FIG. 18) can be communicatively coupled to a secure host tenancy 2004 (e.g., the secure host tenancy 1804 of FIG. 18) that can include a virtual cloud network (VCN) 2006 (e.g., the VCN 1806 of FIG. 18) and a secure host subnet 2008 (e.g., the secure host subnet 1808 of FIG. 18). The VCN 2006 can include an LPG 2010 (e.g., the LPG 1810 of FIG. 18) that can be communicatively coupled to an SSH VCN 2012 (e.g., the SSH VCN 1812 of FIG. 18) via an LPG 2010 contained in the SSH VCN 2012. The SSH VCN 2012 can include an SSH subnet 2014 (e.g., the SSH subnet 1814 of FIG. 18), and the SSH VCN 1812 can be communicatively coupled to a control plane VCN 2016 (e.g. the control plane VCN 1816 of FIG. 18) via an LPG 2010 contained in the control plane VCN 2016 and to a data plane VCN 2018 (e.g. the data plane 1818 of FIG. 18) via an LPG 2010 contained in the data plane VCN 2018. The control plane VCN 2016 and the data plane VCN 2018 can be contained in a service tenancy 2019 (e.g., the service tenancy 1819 of FIG. 18).

**[0270]** The control plane VCN 1816 can include a control plane DMZ tier 1820 (e.g. the control plane DMZ tier 1820 of FIG. 18) that can include load balancer (LB) subnet(s) 1822 (e.g. LB subnet(s) 1822 of FIG. 18), a control plane app tier 2024 (e.g. the control plane app tier 1824 of FIG. 18) that can include app subnet(s) 2026 (e.g., similar to app subnet(s) 1826 of FIG. 18), a control plane data tier 2028 (e.g., the control plane data tier 1828 of FIG. 18) that can include DB subnet(s) 2030. The LB subnet(s) 2022 contained in the control plane DMZ tier 2020 can be communicatively coupled to the app subnet(s) 2026 contained in the control plane app tier 2024 and to an Internet gateway 1834 (e.g. the Internet gateway 1834 of FIG. 18) that can be contained in the control plane VCN 2016, and the app subnet(s) 2026 can

be communicatively coupled to the DB subnet(s) 1830 contained in the control plane data tier 1828 and to a service gateway 1836 (e.g. the service gateway of FIG. 18) and a network address translation (NAT) gateway 1838 (e.g. the NAT gateway 1838 of FIG. 18). The control plane VCN 2016 can include the service gateway 2036 and the NAT gateway 2038.

**[0271]** The data plane VCN 2018 can include a data plane app tier 2046 (e.g., the data plane app tier 1846 of FIG. 18), a data plane DMZ tier 2048 (e.g., the data plane DMZ tier 1848 of FIG. 18), and a data plane data tier 2050 (e.g., the data plane data tier 1850 of FIG. 18). The data plane DMZ tier 2048 can include LB subnet(s) 2022 that can be communicatively coupled to trusted app subnet(s) 2060 and untrusted app subnet(s) 2062 of the data plane app tier 2046 and the Internet gateway 2034 contained in the data plane VCN 2018. The trusted app subnet(s) 2060 can be communicatively coupled to the service gateway 2036 contained in the data plane VCN 2018, the NAT gateway 2038 contained in the data plane VCN 2018, and DB subnet(s) 2030 contained in the data plane data tier 2050. The untrusted app subnet(s) 2062 can be communicatively coupled to the service gateway 2036 contained in the data plane VCN 2018 and DB subnet(s) 2030 contained in the data plane data tier 2050. The data plane data tier 2050 can include DB subnet(s) 2030 that can be communicatively coupled to the service gateway 2036 contained in the data plane VCN 2018.

**[0272]** The untrusted app subnet(s) 2062 can include one or more primary VNICs 2064(1)-(N) that can be communicatively coupled to tenant virtual machines (VMs) 2066(1)-(N). Each tenant VM 2066(1)-(N) can be communicatively coupled to a respective app subnet 2067(1)-(N) that can be contained in respective container egress VCNs 2068(1)-(N) that can be contained in respective customer tenancies 2070(1)-(N). Respective secondary VNICs 2072(1)-(N) can facilitate communication between the untrusted app subnet(s) 2062 contained in the data plane VCN 2018 and the app subnet contained in the container egress VCNs 2068(1)-(N). Each container egress VCNs 2068(1)-(N) can include a NAT gateway 2038 that can be communicatively coupled to public Internet 2054 (e.g., public Internet 1854 of FIG. 18).

**[0273]** The Internet gateway 2034 contained in the control plane VCN 2016 and contained in the data plane VCN 2018 can be communicatively coupled to a metadata management service 2052 (e.g., the metadata management system 1852 of FIG. 18) that can be communicatively coupled to public Internet 2054. Public Internet 2054 can be communicatively coupled to the NAT gateway 2038 contained in the control plane VCN 2016 and contained in the data plane VCN 2018. The service gateway 2036 contained in the control

plane VCN 2016 and contained in the data plane VCN 2018 can be communicatively couple to cloud services 2056.

**[0274]** In some embodiments, the data plane VCN 2018 can be integrated with customer tenancies 2070. This integration can be useful or desirable for customers of the IaaS provider in some cases such as a case that may desire support when executing code. The customer may provide code to run that may be destructive, may communicate with other customer resources, or may otherwise cause undesirable effects. In response to this, the IaaS provider may determine whether to run code given to the IaaS provider by the customer.

**[0275]** In some examples, the customer of the IaaS provider may grant temporary network access to the IaaS provider and request a function to be attached to the data plane tier app 2046. Code to run the function may be executed in the VMs 2066(1)-(N), and the code may not be configured to run anywhere else on the data plane VCN 2018. Each VM 2066(1)-(N) may be connected to one customer tenancy 2070. Respective containers 2071(1)-(N) contained in the VMs 2066(1)-(N) may be configured to run the code. In this case, there can be a dual isolation (e.g., the containers 2071(1)-(N) running code, where the containers 2071(1)-(N) may be contained in at least the VM 2066(1)-(N) that are contained in the untrusted app subnet(s) 2062), which may help prevent incorrect or otherwise undesirable code from damaging the network of the IaaS provider or from damaging a network of a different customer. The containers 2071(1)-(N) may be communicatively coupled to the customer tenancy 2070 and may be configured to transmit or receive data from the customer tenancy 2070. The containers 2071(1)-(N) may not be configured to transmit or receive data from any other entity in the data plane VCN 2018. Upon completion of running the code, the IaaS provider may kill or otherwise dispose of the containers 2071(1)-(N).

**[0276]** In some embodiments, the trusted app subnet(s) 2060 may run code that may be owned or operated by the IaaS provider. In this embodiment, the trusted app subnet(s) 2060 may be communicatively coupled to the DB subnet(s) 2030 and be configured to execute CRUD operations in the DB subnet(s) 2030. The untrusted app subnet(s) 2062 may be communicatively coupled to the DB subnet(s) 2030, but in this embodiment, the untrusted app subnet(s) may be configured to execute read operations in the DB subnet(s) 2030. The containers 2071(1)-(N) that can be contained in the VM 2066(1)-(N) of each customer and that may run code from the customer may not be communicatively coupled with the DB subnet(s) 2030.

**[0277]** In other embodiments, the control plane VCN 2016 and the data plane VCN 2018 may not be directly communicatively coupled. In this embodiment, there may be no direct communication between the control plane VCN 2016 and the data plane VCN 2018. However, communication can occur indirectly through at least one method. An LPG 2010 may be established by the IaaS provider that can facilitate communication between the control plane VCN 2016 and the data plane VCN 2018. In another example, the control plane VCN 2016 or the data plane VCN 2018 can make a call to cloud services 2056 via the service gateway 2036. For example, a call to cloud services 2056 from the control plane VCN 2016 can include a request for a service that can communicate with the data plane VCN 2018.

**[0278]** FIG. 21 is a block diagram 2100 illustrating another example pattern of an IaaS architecture, according to at least one embodiment. Service operators 2102 (e.g., service operators 1802 of FIG. 18) can be communicatively coupled to a secure host tenancy 2104 (e.g., the secure host tenancy 1804 of FIG. 18) that can include a virtual cloud network (VCN) 2106 (e.g., the VCN 1806 of FIG. 18) and a secure host subnet 2108 (e.g., the secure host subnet 1808 of FIG. 18). The VCN 2106 can include an LPG 2110 (e.g., the LPG 1810 of FIG. 18) that can be communicatively coupled to an SSH VCN 2112 (e.g., the SSH VCN 1812 of FIG. 18) via an LPG 2110 contained in the SSH VCN 2112. The SSH VCN 2112 can include an SSH subnet 2114 (e.g., the SSH subnet 1814 of FIG. 18), and the SSH VCN 2112 can be communicatively coupled to a control plane VCN 2116 (e.g., the control plane VCN 1816 of FIG. 18) via an LPG 2110 contained in the control plane VCN 2116 and to a data plane VCN 2118 (e.g., the data plane 1818 of FIG. 18) via an LPG 2110 contained in the data plane VCN 2118. The control plane VCN 2116 and the data plane VCN 2118 can be contained in a service tenancy 2119 (e.g., the service tenancy 1819 of FIG. 18).

**[0279]** The control plane VCN 2116 can include a control plane DMZ tier 2120 (e.g. the control plane DMZ tier 1820 of FIG. 18) that can include LB subnet(s) 2122 (e.g. LB subnet(s) 1822 of FIG. 18), a control plane app tier 2124 (e.g. the control plane app tier 1824 of FIG. 18) that can include app subnet(s) 2126 (e.g. app subnet(s) 1826 of FIG. 18), a control plane data tier 2128 (e.g., the control plane data tier 1828 of FIG. 18) that can include DB subnet(s) 2130 (e.g. DB subnet(s) 2030 of FIG. 20). The LB subnet(s) 2122 contained in the control plane DMZ tier 2120 can be communicatively coupled to the app subnet(s) 2126 contained in the control plane app tier 2124 and to an Internet gateway 2134 (e.g., the Internet gateway 1834 of FIG. 18) that can be contained in the control plane VCN 2116, and the app subnet(s) 2126 can be communicatively coupled to the DB subnet(s) 2130 contained in the control plane data tier

2128 and to a service gateway 2136 (e.g., the service gateway of FIG. 18) and a network address translation (NAT) gateway 2138 (e.g., the NAT gateway 1838 of FIG. 18). The control plane VCN 2116 can include the service gateway 2136 and the NAT gateway 2138.

**[0280]** The data plane VCN 2118 can include a data plane app tier 2146 (e.g., the data plane app tier 1846 of FIG. 18), a data plane DMZ tier 2148 (e.g., the data plane DMZ tier 2148 of FIG. 18), and a data plane data tier 2150 (e.g., the data plane data tier 1850 of FIG. 18). The data plane DMZ tier 2148 can include LB subnet(s) 2122 that can be communicatively coupled to trusted app subnet(s) 2160 (e.g., trusted app subnet(s) 2060 of FIG. 20) and untrusted app subnet(s) 2162 (e.g., untrusted app subnet(s) 2062 of FIG. 20) of the data plane app tier 2146 and the Internet gateway 2134 contained in the data plane VCN 2118. The trusted app subnet(s) 2160 can be communicatively coupled to the service gateway 2136 contained in the data plane VCN 2118, the NAT gateway 2138 contained in the data plane VCN 2118, and DB subnet(s) 2130 contained in the data plane data tier 2150. The untrusted app subnet(s) 2162 can be communicatively coupled to the service gateway 2136 contained in the data plane VCN 2118 and DB subnet(s) 2130 contained in the data plane data tier 2150. The data plane data tier 2150 can include DB subnet(s) 2130 that can be communicatively coupled to the service gateway 2136 contained in the data plane VCN 2118.

**[0281]** The untrusted app subnet(s) 2162 can include primary VNICs 2164(1)-(N) that can be communicatively coupled to tenant virtual machines (VMs) 2166(1)-(N) residing within the untrusted app subnet(s) 2162. Each tenant VM 2166(1)-(N) can run code in a respective container 2167(1)-(N) and be communicatively coupled to an app subnet 2126 that can be contained in a data plane app tier 2146 that can be contained in a container egress VCN 2168. Respective secondary VNICs 2172(1)-(N) can facilitate communication between the untrusted app subnet(s) 2162 contained in the data plane VCN 2118 and the app subnet contained in the container egress VCN 2168. The container egress VCN can include a NAT gateway 2138 that can be communicatively coupled to public Internet 2154 (e.g., public Internet 1854 of FIG. 18).

**[0282]** The Internet gateway 2134 contained in the control plane VCN 2116 and contained in the data plane VCN 2118 can be communicatively coupled to a metadata management service 2152 (e.g., the metadata management system 1852 of FIG. 18) that can be communicatively coupled to public Internet 2154. Public Internet 2154 can be communicatively coupled to the NAT gateway 2138 contained in the control plane VCN 2116 and contained in the data plane VCN 2118. The service gateway 2136 contained in the control

plane VCN 2116 and contained in the data plane VCN 2118 can be communicatively couple to cloud services 2156.

**[0283]** In some examples, the pattern illustrated by the architecture of block diagram 2100 of FIG. 21 may be considered an exception to the pattern illustrated by the architecture of block diagram 2000 of FIG. 20 and may be desirable for a customer of the IaaS provider if the IaaS provider cannot directly communicate with the customer (e.g., a disconnected region). The respective containers 2167(1)-(N) that are contained in the VMs 2166(1)-(N) for each customer can be accessed in real-time by the customer. The containers 2167(1)-(N) may be configured to make calls to respective secondary VNICs 2172(1)-(N) contained in app subnet(s) 2126 of the data plane app tier 2146 that can be contained in the container egress VCN 2168. The secondary VNICs 2172(1)-(N) can transmit the calls to the NAT gateway 2138 that may transmit the calls to public Internet 2154. In this example, the containers 2167(1)-(N) that can be accessed in real-time by the customer can be isolated from the control plane VCN 2116 and can be isolated from other entities contained in the data plane VCN 2118. The containers 2167(1)-(N) may also be isolated from resources from other customers.

**[0284]** In other examples, the customer can use the containers 2167(1)-(N) to call cloud services 2156. In this example, the customer may run code in the containers 2167(1)-(N) that requests a service from cloud services 2156. The containers 2167(1)-(N) can transmit this request to the secondary VNICs 2172(1)-(N) that can transmit the request to the NAT gateway that can transmit the request to public Internet 2154. Public Internet 2154 can transmit the request to LB subnet(s) 2122 contained in the control plane VCN 2116 via the Internet gateway 2134. In response to determining the request is valid, the LB subnet(s) can transmit the request to app subnet(s) 2126 that can transmit the request to cloud services 2156 via the service gateway 2136.

**[0285]** It should be appreciated that IaaS architectures 1800, 1900, 2000, 2100 depicted in the figures may have other components than those depicted. Further, the embodiments shown in the figures are only some examples of a cloud infrastructure system that may incorporate an embodiment of the disclosure. In some other embodiments, the IaaS systems may have more or fewer components than shown in the figures, may combine two or more components, or may have a different configuration or arrangement of components.

**[0286]** In certain embodiments, the IaaS systems described herein may include a suite of applications, middleware, and database service offerings that are delivered to a customer in a self-service, subscription-based, elastically scalable, reliable, highly available, and secure

manner. An example of such an IaaS system is the Oracle Cloud Infrastructure (OCI) provided by the present assignee.

**[0287]** FIG. 22 illustrates an example computer system 2200, in which various embodiments of the present disclosure may be implemented. The system 2200 may be used to implement any of the computer systems described above. As shown in the figure, computer system 2200 includes a processing unit 2204 that communicates with a number of peripheral subsystems via a bus subsystem 2202. These peripheral subsystems may include a processing acceleration unit 2206, an I/O subsystem 2208, a storage subsystem 2218 and a communications subsystem 2224. Storage subsystem 2218 includes tangible computer-readable storage media 2222 and a system memory 2210.

**[0288]** Bus subsystem 2202 provides a mechanism for letting the various components and subsystems of computer system 2200 communicate with each other as intended. Although bus subsystem 2202 is shown schematically as a single bus, alternative embodiments of the bus subsystem may utilize multiple buses. Bus subsystem 2202 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. For example, such architectures may include an Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus, which can be implemented as a Mezzanine bus manufactured to the IEEE P1386.1 standard.

**[0289]** Processing unit 2204, which can be implemented as one or more integrated circuits (e.g., a conventional microprocessor or microcontroller), controls the operation of computer system 2200. One or more processors may be included in processing unit 2204. These processors may include single core or multicore processors. In certain embodiments, processing unit 2204 may be implemented as one or more independent processing units 2232 and/or 2234 with single or multicore processors included in each processing unit. In other embodiments, processing unit 2204 may also be implemented as a quad-core processing unit formed by integrating two dual-core processors into a single chip.

**[0290]** In various embodiments, processing unit 2204 can execute a variety of programs in response to program code and can maintain multiple concurrently executing programs or processes. At any given time, some, or all of the program code to be executed can be resident in processor(s) 2204 and/or in storage subsystem 2218. Through suitable programming, processor(s) 2204 can provide various functionalities described above. Computer system 2200

may additionally include a processing acceleration unit 2206, which can include a digital signal processor (DSP), a special-purpose processor, and/or the like.

**[0291]** I/O subsystem 2208 may include user interface input devices and user interface output devices. User interface input devices may include a keyboard, pointing devices such as a mouse or trackball, a touchpad or touch screen incorporated into a display, a scroll wheel, a click wheel, a dial, a button, a switch, a keypad, audio input devices with voice command recognition systems, microphones, and other types of input devices. User interface input devices may include, for example, motion sensing and/or gesture recognition devices such as the Microsoft Kinect® motion sensor that enables users to control and interact with an input device, such as the Microsoft Xbox® 360 game controller, through a natural user interface using gestures and spoken commands. User interface input devices may also include eye gesture recognition devices such as the Google Glass® blink detector that detects eye activity (e.g., 'blinking' while taking pictures and/or making a menu selection) from users and transforms the eye gestures as input into an input device (e.g., Google Glass®). Additionally, user interface input devices may include voice recognition sensing devices that enable users to interact with voice recognition systems (e.g., Siri® navigator), through voice commands.

**[0292]** User interface input devices may also include, without limitation, three dimensional (3D) mice, joysticks or pointing sticks, gamepads and graphic tablets, and audio/visual devices such as speakers, digital cameras, digital camcorders, portable media players, webcams, image scanners, fingerprint scanners, barcode reader 3D scanners, 3D printers, laser rangefinders, and eye gaze tracking devices. Additionally, user interface input devices may include, for example, medical imaging input devices such as computed tomography, magnetic resonance imaging, position emission tomography, medical ultrasonography devices. User interface input devices may also include, for example, audio input devices such as MIDI keyboards, digital musical instruments and the like.

**[0293]** User interface output devices may include a display subsystem, indicator lights, or non-visual displays such as audio output devices, etc. The display subsystem may be a cathode ray tube (CRT), a flat-panel device, such as that using a liquid crystal display (LCD) or plasma display, a projection device, a touch screen, and the like. In general, use of the term "output device" is intended to include all possible types of devices and mechanisms for outputting information from computer system 2200 to a user or other computer. For example, user interface output devices may include, without limitation, a variety of display devices that visually convey text, graphics, and audio/video information such as monitors, printers,

speakers, headphones, automotive navigation systems, plotters, voice output devices, and modems.

**[0294]** Computer system 2200 may comprise a storage subsystem 2218 that comprises software elements, shown as being currently located within a system memory 2210. System memory 2210 may store program instructions that are loadable and executable on processing unit 2204, as well as data generated during the execution of these programs.

**[0295]** Depending on the configuration and type of computer system 2200, system memory 2210 may be volatile (such as random-access memory (RAM)) and/or non-volatile (such as read-only memory (ROM), flash memory, etc.) The RAM typically contains data and/or program modules that are immediately accessible to and/or presently being operated and executed by processing unit 2204. In some implementations, system memory 2210 may include multiple different types of memory, such as static random-access memory (SRAM) or dynamic random-access memory (DRAM). In some implementations, a basic input/output system (BIOS), containing the basic routines that help to transfer information between elements within computer system 2200, such as during start-up, may typically be stored in the ROM. By way of example, and not limitation, system memory 2210 also illustrates application programs 2212, which may include client applications, Web browsers, mid-tier applications, relational database management systems (RDBMS), etc., program data 2214, and an operating system 2216. By way of example, operating system 2216 may include various versions of Microsoft Windows®, Apple Macintosh®, and/or Linux operating systems, a variety of commercially available UNIX® or UNIX-like operating systems (including without limitation the variety of GNU/Linux operating systems, the Google Chrome® OS, and the like) and/or mobile operating systems such as iOS, Windows® Phone, Android® OS, BlackBerry® 17 OS, and Palm® OS operating systems.

**[0296]** Storage subsystem 2218 may also provide a tangible computer-readable storage medium for storing the basic programming and data constructs that provide the functionality of some embodiments. Software (programs, code modules, instructions) that when executed by a processor provide the functionality described above may be stored in storage subsystem 2218. These software modules or instructions may be executed by processing unit 2204. Storage subsystem 2218 may also provide a repository for storing data used in accordance with the present disclosure.

**[0297]** Storage subsystem 2200 may also include a computer-readable storage media reader 2220 that can further be connected to computer-readable storage media 2222. Together and

optionally, in combination with system memory 2210, computer-readable storage media 2222 may comprehensively represent remote, local, fixed, and/or removable storage devices plus storage media for temporarily and/or more permanently containing, storing, transmitting, and retrieving computer-readable information.

**[0298]** Computer-readable storage media 2222 containing code, or portions of code, can also include any appropriate media known or used in the art, including storage media and communication media, such as but not limited to, volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage and/or transmission of information. This can include tangible computer-readable storage media such as RAM, ROM, electronically erasable programmable ROM (EEPROM), flash memory or other memory technology, CD-ROM, digital versatile disk (DVD), or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or other tangible computer readable media. This can also include nontangible computer-readable media, such as data signals, data transmissions, or any other medium which can be used to transmit the desired information, and which can be accessed by computing system 2200.

**[0299]** By way of example, computer-readable storage media 2222 may include a hard disk drive that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive that reads from or writes to a removable, nonvolatile magnetic disk, and an optical disk drive that reads from or writes to a removable, nonvolatile optical disk such as a CD ROM, DVD, and Blu-Ray® disk, or other optical media. Computer-readable storage media 2222 may include, but is not limited to, Zip® drives, flash memory cards, universal serial bus (USB) flash drives, secure digital (SD) cards, DVD disks, digital video tape, and the like. Computer-readable storage media 2222 may also include, solid-state drives (SSD) based on non-volatile memory such as flash-memory based SSDs, enterprise flash drives, solid state ROM, and the like, SSDs based on volatile memory such as solid-state RAM, dynamic RAM, static RAM, DRAM-based SSDs, magneto resistive RAM (MRAM) SSDs, and hybrid SSDs that use a combination of DRAM and flash memory based SSDs. The disk drives and their associated computer-readable media may provide non-volatile storage of computer-readable instructions, data structures, program modules, and other data for computer system 2200.

**[0300]** Communications subsystem 2224 provides an interface to other computer systems and networks. Communications subsystem 2224 serves as an interface for receiving data from and transmitting data to other systems from computer system 2200. For example, communications subsystem 2224 may enable computer system 2200 to connect to one or more

devices via the Internet. In some embodiments communications subsystem 2224 can include radio frequency (RF) transceiver components for accessing wireless voice and/or data networks (e.g., using cellular telephone technology, advanced data network technology, such as 3G, 4G or EDGE (enhanced data rates for global evolution), Wi-Fi (IEEE 802.11 family standards, or other mobile communication technologies, or any combination thereof), global positioning system (GPS) receiver components, and/or other components. In some embodiments communications subsystem 2224 can provide wired network connectivity (e.g., Ethernet) in addition to or instead of a wireless interface.

**[0301]** In some embodiments, communications subsystem 2224 may also receive input communication in the form of structured and/or unstructured data feeds 2226, event streams 2228, event updates 2230, and the like on behalf of one or more users who may use computer system 2200.

**[0302]** By way of example, communications subsystem 2224 may be configured to receive data feeds 2226 in real-time from users of social networks and/or other communication services such as Twitter® feeds, Facebook® updates, web feeds such as Rich Site Summary (RSS) feeds, and/or real-time updates from one or more third party information sources.

**[0303]** Additionally, communications subsystem 2224 may also be configured to receive data in the form of continuous data streams, which may include event streams 2228 of real-time events and/or event updates 2230 that may be continuous or unbounded in nature with no explicit end. Examples of applications that generate continuous data may include, for example, sensor data applications, financial tickers, network performance measuring tools (e.g., network monitoring and traffic management applications), clickstream analysis tools, automobile traffic monitoring, and the like.

**[0304]** Communications subsystem 2224 may also be configured to output the structured and/or unstructured data feeds 2226, event streams 2228, event updates 2230, and the like to one or more databases that may be in communication with one or more streaming data source computers coupled to computer system 2200.

**[0305]** Computer system 2200 can be one of various types, including a handheld portable device (e.g., an iPhone® cellular phone, an iPad® computing tablet, a PDA), a wearable device (e.g., a Google Glass® head mounted display), a PC, a workstation, a mainframe, a kiosk, a server rack, or any other data processing system.

**[0306]** Due to the ever-changing nature of computers and networks, the description of computer system 2200 depicted in the figure is intended only as a specific example. Many other

configurations having more or fewer components than the system depicted in the figure are possible. For example, customized hardware might also be used and/or particular elements might be implemented in hardware, firmware, software (including applets), or a combination. Further, connection to other computing devices, such as network input/output devices, may be employed. Based on the disclosure and teachings provided herein, a person of ordinary skill in the art will appreciate other ways and/or methods to implement the various embodiments.

**[0307]** Although specific embodiments of the disclosure have been described, various modifications, alterations, alternative constructions, and equivalents are also encompassed within the scope of the disclosure. Embodiments of the present disclosure are not restricted to operation within certain specific data processing environments but are free to operate within a plurality of data processing environments. Additionally, although embodiments of the present disclosure have been described using a particular series of transactions and steps, it should be apparent to those skilled in the art that the scope of the present disclosure is not limited to the described series of transactions and steps. Various features and aspects of the above-described embodiments may be used individually or jointly.

**[0308]** Further, while embodiments of the present disclosure have been described using a particular combination of hardware and software, it should be recognized that other combinations of hardware and software are also within the scope of the present disclosure. Embodiments of the present disclosure may be implemented only in hardware, or only in software, or using combinations thereof. The various processes described herein can be implemented on the same processor or different processors in any combination. Accordingly, where components or modules are described as being configured to perform certain operations, such configuration can be accomplished, e.g., by designing electronic circuits to perform the operation, by programming programmable electronic circuits (such as microprocessors) to perform the operation, or any combination thereof. Processes can communicate using a variety of techniques including but not limited to conventional techniques for inter process communication, and different pairs of processes may use different techniques, or the same pair of processes may use different techniques at different times.

**[0309]** The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will, however, be evident that additions, subtractions, deletions, and other modifications and changes may be made thereunto without departing from the broader spirit and scope as set forth in the claims. Thus, although specific disclosure

embodiments have been described, these are not intended to be limiting. Various modifications and equivalents are within the scope of the following claims.

**[0310]** The use of the terms "a" and "an" and "the" and similar referents in the context of describing the disclosed embodiments (especially in the context of the following claims) are to be construed to cover both the singular and the plural, unless otherwise indicated herein or clearly contradicted by context. The terms "comprising," "having," "including," and "containing" are to be construed as open-ended terms (i.e., meaning "including, but not limited to,") unless otherwise noted. The term "connected" is to be construed as partly or wholly contained within, attached to, or joined together, even if there is something intervening. Recitation of ranges of values herein are merely intended to serve as a shorthand method of referring individually to each separate value falling within the range, unless otherwise indicated herein and each separate value is incorporated into the specification as if it were individually recited herein. All methods described herein can be performed in any suitable order unless otherwise indicated herein or otherwise clearly contradicted by context. The use of any and all examples, or exemplary language (e.g., "such as") provided herein, is intended merely to better illuminate embodiments of the disclosure and does not pose a limitation on the scope of the disclosure unless otherwise claimed. No language in the specification should be construed as indicating any non-claimed element as essential to the practice of the disclosure.

**[0311]** Disjunctive language such as the phrase "at least one of X, Y, or Z," unless specifically stated otherwise, is intended to be understood within the context as used in general to present that an item, term, etc., may be either X, Y, or Z, or any combination thereof (e.g., X, Y, and/or Z). Thus, such disjunctive language is not generally intended to, and should not, imply that certain embodiments require at least one of X, at least one of Y, or at least one of Z to each be present.

**[0312]** Preferred embodiments of this disclosure are described herein, including the best mode known to the inventors for carrying out the disclosure. Variations of those preferred embodiments may become apparent to those of ordinary skill in the art upon reading the foregoing description. The inventors expect skilled artisans to employ such variations as appropriate and the inventors intend for the disclosure to be practiced otherwise than as specifically described herein. Accordingly, this disclosure includes all modifications and equivalents of the subject matter recited in the claims appended hereto as permitted by applicable law. Moreover, any combination of the above-described elements in all possible

variations thereof is encompassed by the disclosure unless otherwise indicated herein or otherwise clearly contradicted by context.

**[0313]** All references, including publications, patent applications, and patents, cited herein are hereby incorporated by reference to the same extent as if each reference were individually and specifically indicated to be incorporated by reference and were set forth in its entirety herein.

In the foregoing specification, aspects of the disclosure are described with reference to specific embodiments thereof, but those skilled in the art will recognize that the disclosure is not limited thereto. Various features and aspects of the above-described disclosure may be used individually or jointly. Further, embodiments can be utilized in any number of environments and applications beyond those described herein without departing from the broader spirit and scope of the specification. The specification and drawings are, accordingly, to be regarded as illustrative rather than restrictive.

WHAT IS CLAIMED IS:

1. A method comprising:

receiving, by a first cloud infrastructure provided by a first cloud services provider (CSP), a request from a user associated with an account in a second cloud infrastructure provided by a second CSP, the request requesting use of a service provided by the first cloud infrastructure;

creating a tenancy for the user in the first cloud infrastructure to enable the user to utilize the service; and

creating, by the first cloud infrastructure, a link-resource object that includes information linking the tenancy of the user in the first cloud infrastructure to the account of the user in the second cloud infrastructure, the link-resource object enabling the user to utilize the service provided by the first cloud infrastructure.

2. The method of claim 1, further comprising:

validating, by the first cloud infrastructure, the user based on a token included in the request, wherein the tenancy for the user in the first cloud infrastructure is created upon the user being successfully validated.

3. The method of claim 2, wherein the token includes information identifying a role associated with the user in the second cloud infrastructure, and wherein the role is associated with one or more tasks/operations that are permitted for the role.

4. The method of claim 2, wherein the validating includes:

obtaining a role associated with the user in the second cloud infrastructure based on the token;

identifying one or more tasks/operations associated with the role; and

determining, based on the one or more tasks/operations, whether the user is permitted to issue the request.

5. The method of claim 1, wherein linking the tenancy of the user in the first cloud infrastructure to the account of the user in the second cloud infrastructure enables the user to utilize the service provided by the first cloud infrastructure using identity information of the user associated with the second cloud infrastructure.

6. The method of claim 1, wherein linking the tenancy of the user in the first cloud infrastructure to the account of the user in the second cloud infrastructure includes storing, in the link-resource object, a mapping of a first identifier associated with the tenancy of the user in the first cloud infrastructure to a second identifier associated with the account of the user in the second cloud infrastructure.

7. The method of claim 1, further comprising:

creating, by the first cloud infrastructure, a resource-principal that is associated with the link-resource object, the resource-principal being assigned one or more permissions based on a token included in the request.

8. The method of claim 7, wherein the resource-principal and the link-resource object are stored in a root compartment of the tenancy of the user in the first cloud infrastructure.

9. The method of claim 1, further comprising:

federating the account of the user in the second cloud infrastructure with the tenancy of the user in the first cloud infrastructure, wherein the federating includes synchronizing users/groups information included in the account with users/groups information included in the tenancy.

10. The method of claim 1, wherein the first cloud infrastructure receives the request from a multi-cloud console.

11. One or more computer readable non-transitory media storing computer-executable instructions that, when executed by one or more processors, cause:

receiving, by a first cloud infrastructure provided by a first cloud services provider (CSP), a request from a user associated with an account in a second cloud infrastructure provided by a second CSP, the request requesting use of a service provided by the first cloud infrastructure;

creating a tenancy for the user in the first cloud infrastructure to enable the user to utilize the service; and

creating, by the first cloud infrastructure, a link-resource object that includes information linking the tenancy of the user in the first cloud infrastructure to the account of the user in the

second cloud infrastructure, the link-resource object enabling the user to utilize the service provided by the first cloud infrastructure.

12. The one or more computer readable non-transitory media storing computer-executable instructions of claim 11, further comprising instructions that, when executed by one or more processors, cause:

validating, by the first cloud infrastructure, the user based on a token included in the request requesting use of the service provided by the first cloud infrastructure, wherein the tenancy for the user in the first cloud infrastructure is created upon the user being successfully validated.

13. The one or more computer readable non-transitory media storing computer-executable instructions of claim 12, wherein the token includes information identifying a role associated with the user in the second cloud infrastructure, and wherein the role is associated with one or more tasks/operations that are permitted for the role.

14. The one or more computer readable non-transitory media storing computer-executable instructions of claim 12, wherein the validating includes:

obtaining a role associated with the user in the second cloud infrastructure based on the token;

identifying one or more tasks/operations associated with the role; and

determining, based on the one or more tasks/operations, whether the user is permitted to issue the request requesting use of the service provided by the first cloud infrastructure.

15. The one or more computer readable non-transitory media storing computer-executable instructions of claim 11, wherein linking the tenancy of the user in the first cloud infrastructure to the account of the user in the second cloud infrastructure enables the user to utilize the service provided by the first cloud infrastructure using identity information of the user associated with the second cloud infrastructure.

16. The one or more computer readable non-transitory media storing computer-executable instructions of claim 11, wherein linking the tenancy of the user in the first cloud infrastructure to the account of the user in the second cloud infrastructure includes storing, in

the link-resource object, a mapping of a first identifier associated with the tenancy of the user in the first cloud infrastructure to a second identifier associated with the account of the user in the second cloud infrastructure.

17. The one or more computer readable non-transitory media storing computer-executable instructions of claim 11, further comprising instructions that, when executed by one or more processors, cause:

creating, by the first cloud infrastructure, a resource-principal that is associated with the link-resource object, the resource-principal being assigned one or more permissions based on a token included in the request requesting use of the service provided by the first cloud infrastructure.

18. The one or more computer readable non-transitory media storing computer-executable instructions of claim 17, wherein the resource-principal and the link-resource object are stored in a root compartment of the tenancy of the user in the first cloud infrastructure.

19. The one or more computer readable non-transitory media storing computer-executable instructions of claim 11, wherein the first cloud infrastructure receives the request requesting use of the service provided by the first cloud infrastructure from a multi-cloud console.

20. A computing device comprising:

one or more processors; and

a memory including instructions that, when executed with the one or more processors, cause the computing device to, at least:

receive, by a first cloud infrastructure provided by a first cloud services provider (CSP), a request from a user associated with an account in a second cloud infrastructure provided by a second CSP, the request requesting use of a service provided by the first cloud infrastructure;

create a tenancy for the user in the first cloud infrastructure to enable the user to utilize the service; and

create, by the first cloud infrastructure, a link-resource object that includes information linking the tenancy of the user in the first cloud infrastructure to the account of the

user in the second cloud infrastructure, the link-resource object enabling the user to utilize the service provided by the first cloud infrastructure.

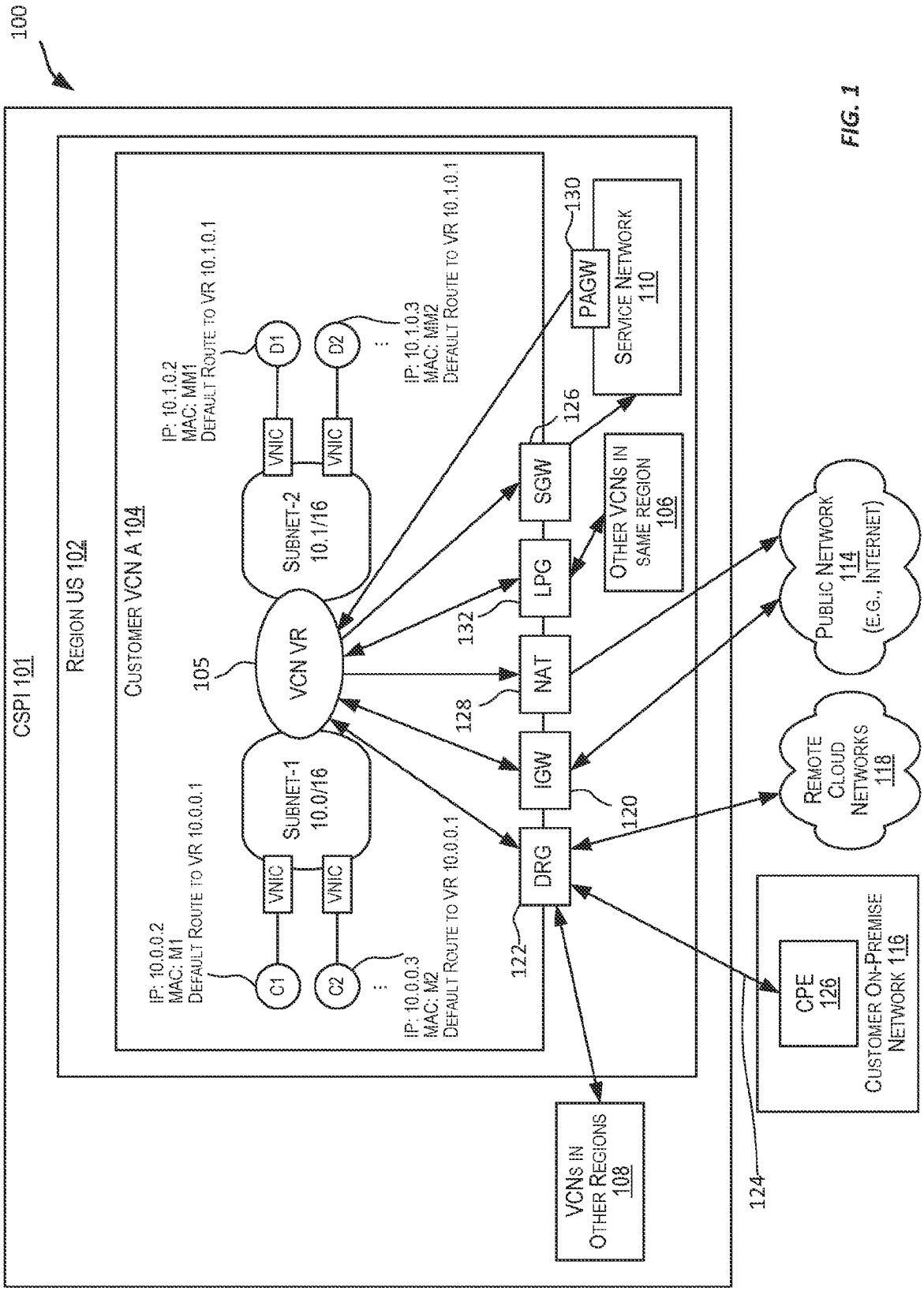
100
CSPI 101
REGION US 102
CUSTOMER VCN A 104
105
VCN VR
SUBNET-1
10.0/16
SUBNET-2
10.1/16
VNIC
C1
C2
D1
D2
IP: 10.0.0.2
MAC: M1
DEFAULT ROUTE TO VR 10.0.0.1
IP: 10.0.0.3
MAC: M2
DEFAULT ROUTE TO VR 10.0.0.1
IP: 10.1.0.2
MAC: MM1
DEFAULT ROUTE TO VR 10.1.0.1
IP: 10.1.0.3
MAC: MM2
DEFAULT ROUTE TO VR 10.1.0.1
DRG
IGW
NAT
LPG
SGW
PAGW
120
122
124
126
128
130
132
SERVICE NETWORK 110
OTHER VCNS IN SAME REGION 106
PUBLIC NETWORK 114 (E.G., INTERNET)
REMOTE CLOUD NETWORKS 118
CPE 126
CUSTOMER ON-PREMISE NETWORK 116
VCNS IN OTHER REGIONS 108

FIG. 1

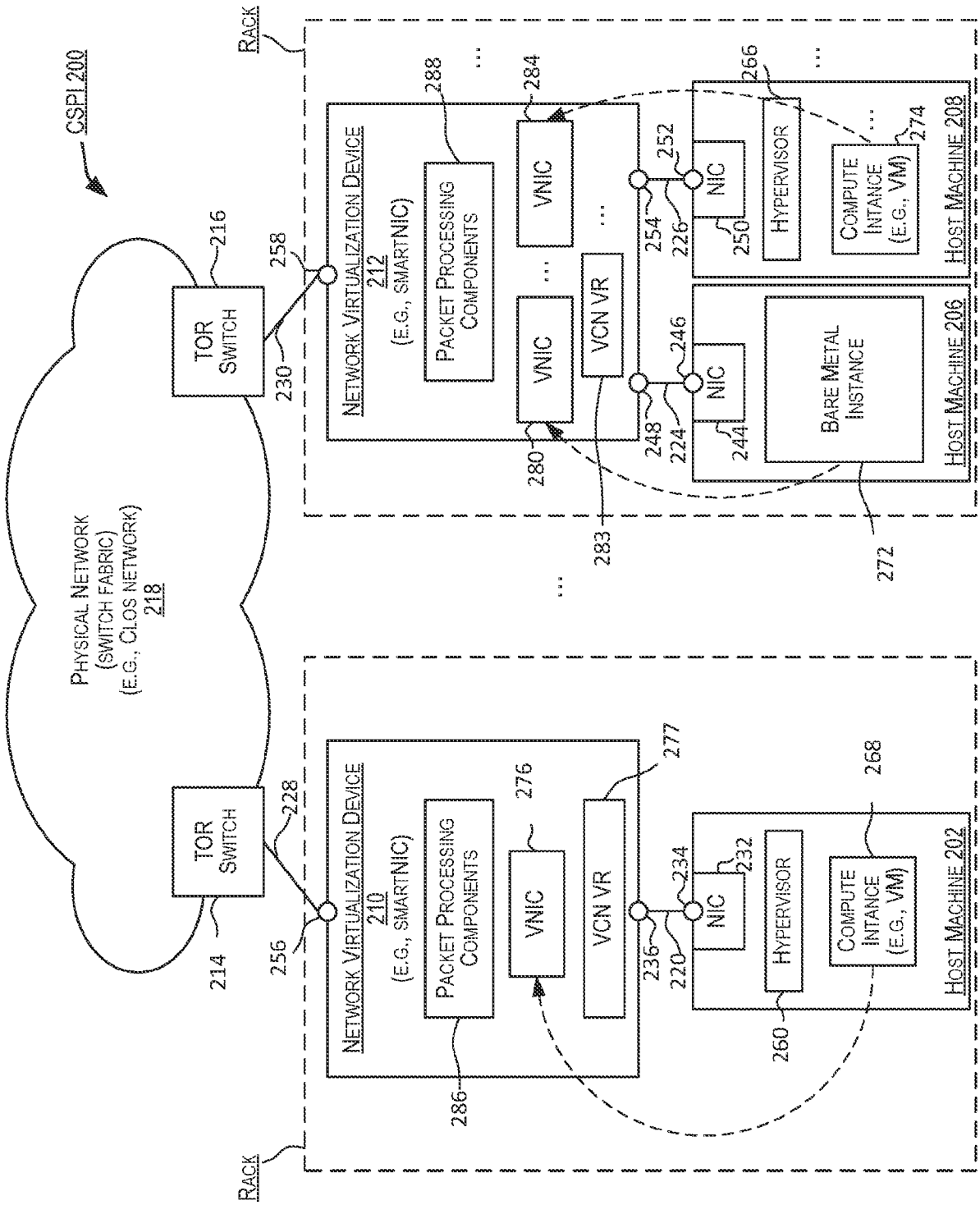
CSPI 200
PHYSICAL NETWORK (SWITCH FABRIC) (E.G., CLOS NETWORK) 218
TOR SWITCH
216
214
TOR SWITCH
228
230
258
256
RACK
NETWORK VIRTUALIZATION DEVICE 212 (E.G., SMARTNIC)
PACKET PROCESSING COMPONENTS
288
VNIC
284
VCN VR
283
VNIC
280
NIC
252
254
226
250
HYPERVISOR
266
COMPUTE INTANCE (E.G., VM)
274
HOST MACHINE 208
NIC
246
248
224
244
BARE METAL INSTANCE
272
HOST MACHINE 206
RACK
NETWORK VIRTUALIZATION DEVICE 210 (E.G., SMARTNIC)
PACKET PROCESSING COMPONENTS
286
VNIC
276
VCN VR
277
NIC
234
236
220
232
HYPERVISOR
260
COMPUTE INTANCE (E.G., VM)
268
HOST MACHINE 202

FIG. 2

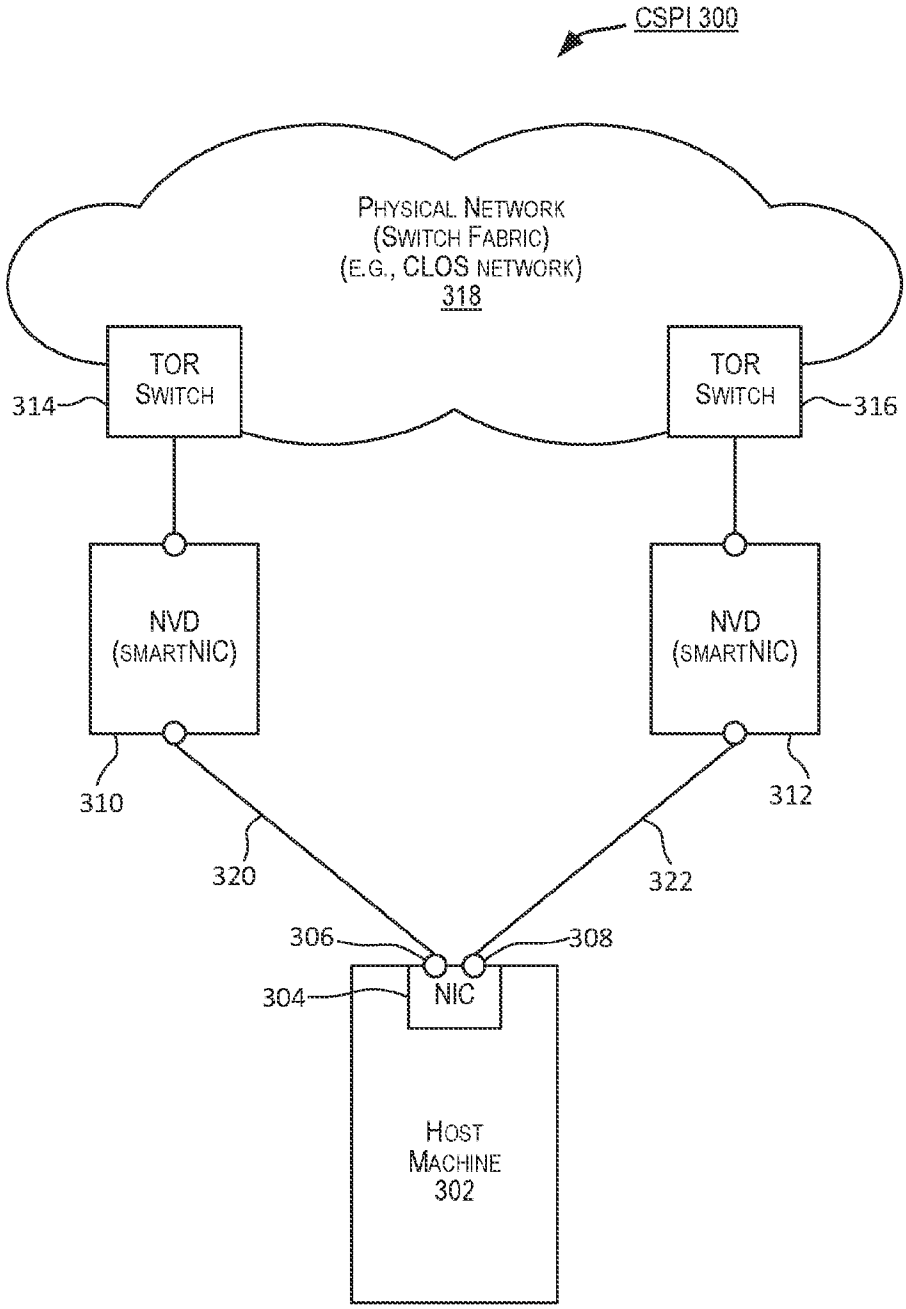
CSPI 300
PHYSICAL NETWORK
(SWITCH FABRIC)
(E.G., CLOS NETWORK)
318
TOR SWITCH
314
TOR SWITCH
316
NVD (SMARTNIC)
NVD (SMARTNIC)
310
312
320
322
306
308
304
NIC
HOST MACHINE
302

*FIG. 3*

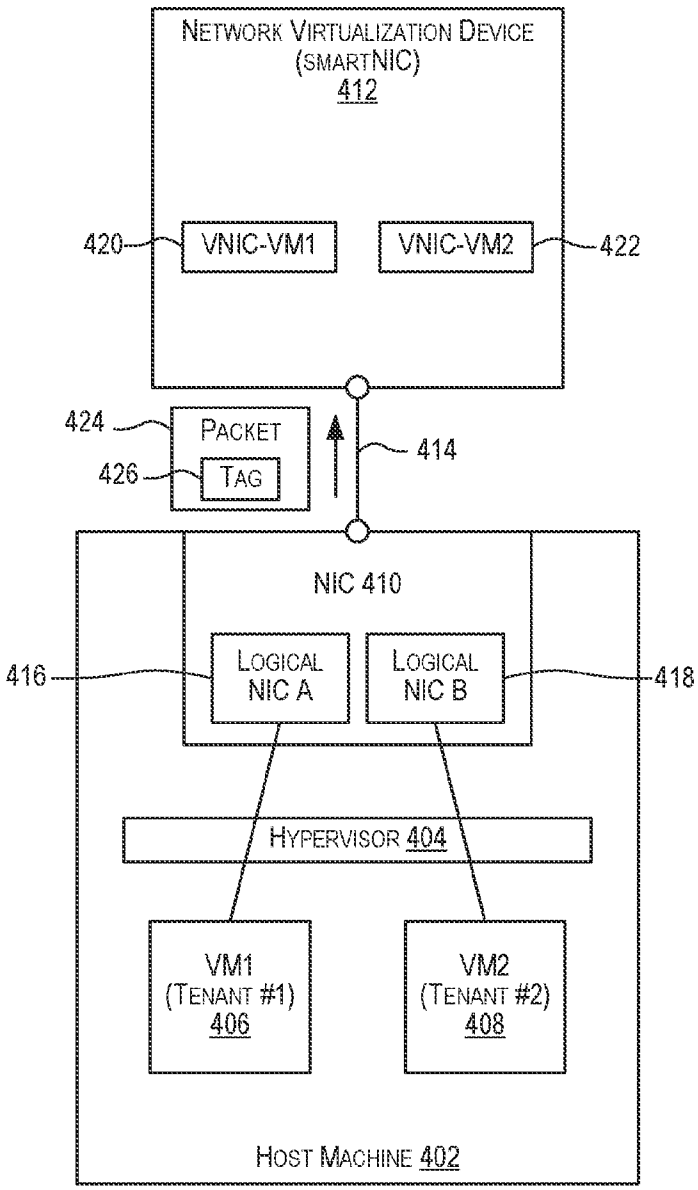
NETWORK VIRTUALIZATION DEVICE
(SMARTNIC)
412
420
VNIC-VM1
VNIC-VM2
422
424
PACKET
TAG
426
414
NIC 410
416
LOGICAL
NIC A
LOGICAL
NIC B
418
HYPERVISOR 404
VM1
(TENANT #1)
406
VM2
(TENANT #2)
408
HOST MACHINE 402

*FIG. 4*

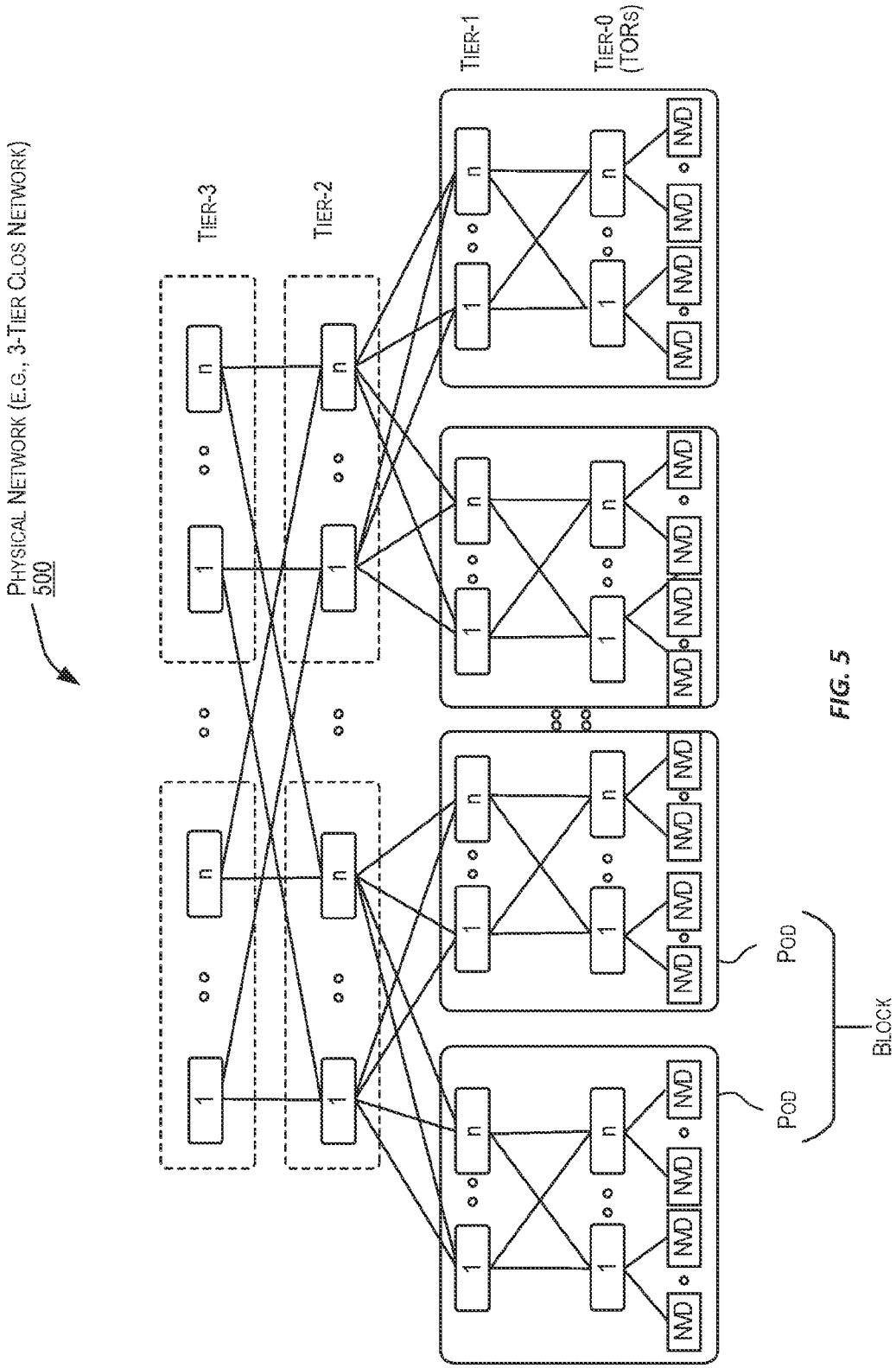
PHYSICAL NETWORK (E.G., 3-TIER CLOS NETWORK)
500
TIER-3
TIER-2
TIER-1
TIER-0 (TORS)
n
1
NMD
POD
POD
BLOCK

FIG. 5

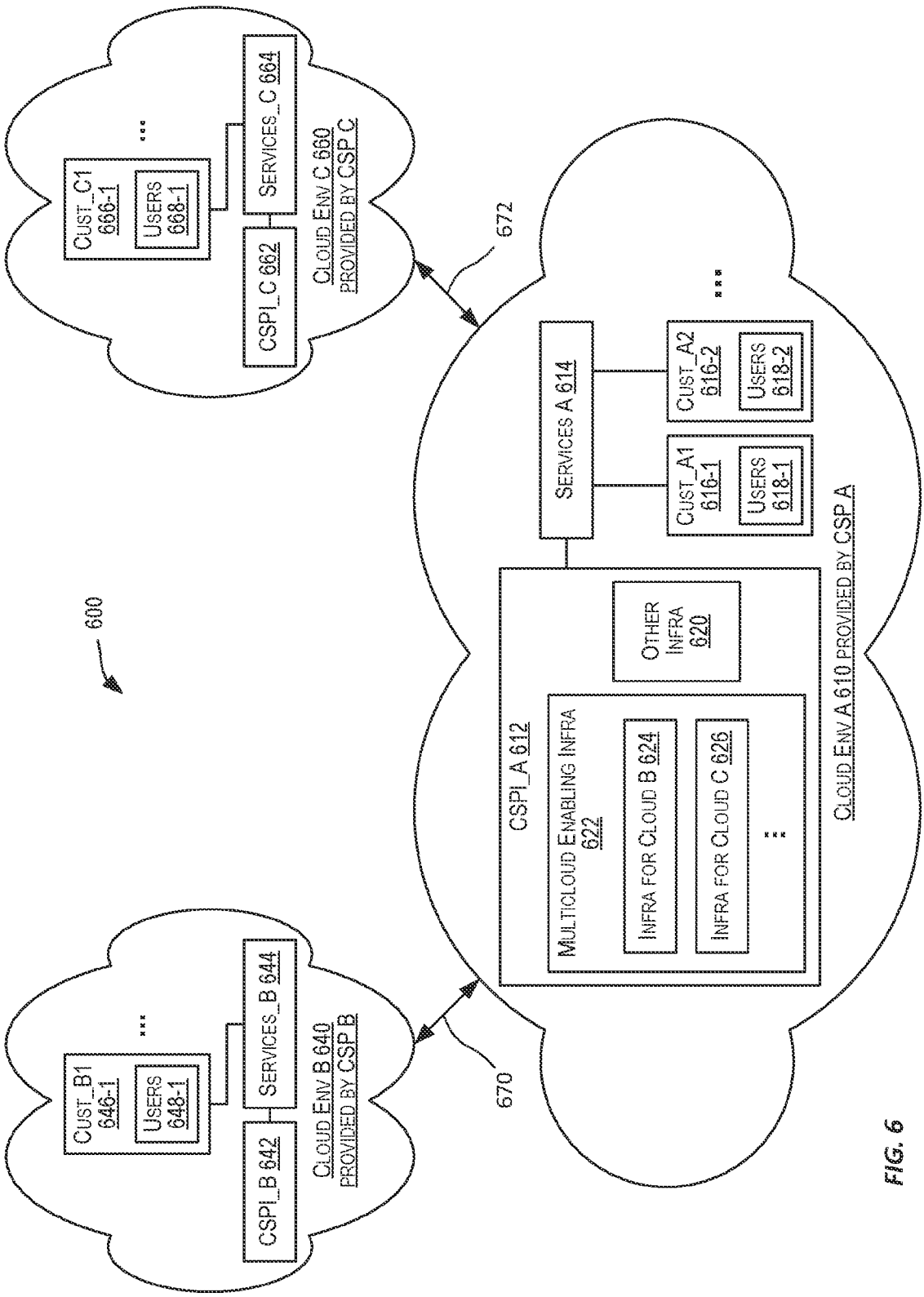
600
CUST_B1 646-1
USERS 648-1
...
CSPI_B 642
SERVICES_B 644
CLOUD ENV B 640 PROVIDED BY CSP B
670
672
CUST_C1 666-1
USERS 668-1
...
CSPI_C 662
SERVICES_C 664
CLOUD ENV C 660 PROVIDED BY CSP C
CSPI_A 612
MULTICLOUD ENABLING INFRA 622
INFRA FOR CLOUD B 624
INFRA FOR CLOUD C 626
...
OTHER INFRA 620
SERVICES A 614
CUST_A1 616-1
USERS 618-1
CUST_A2 616-2
USERS 618-2
...
CLOUD ENV A 610 PROVIDED BY CSP A

FIG. 6

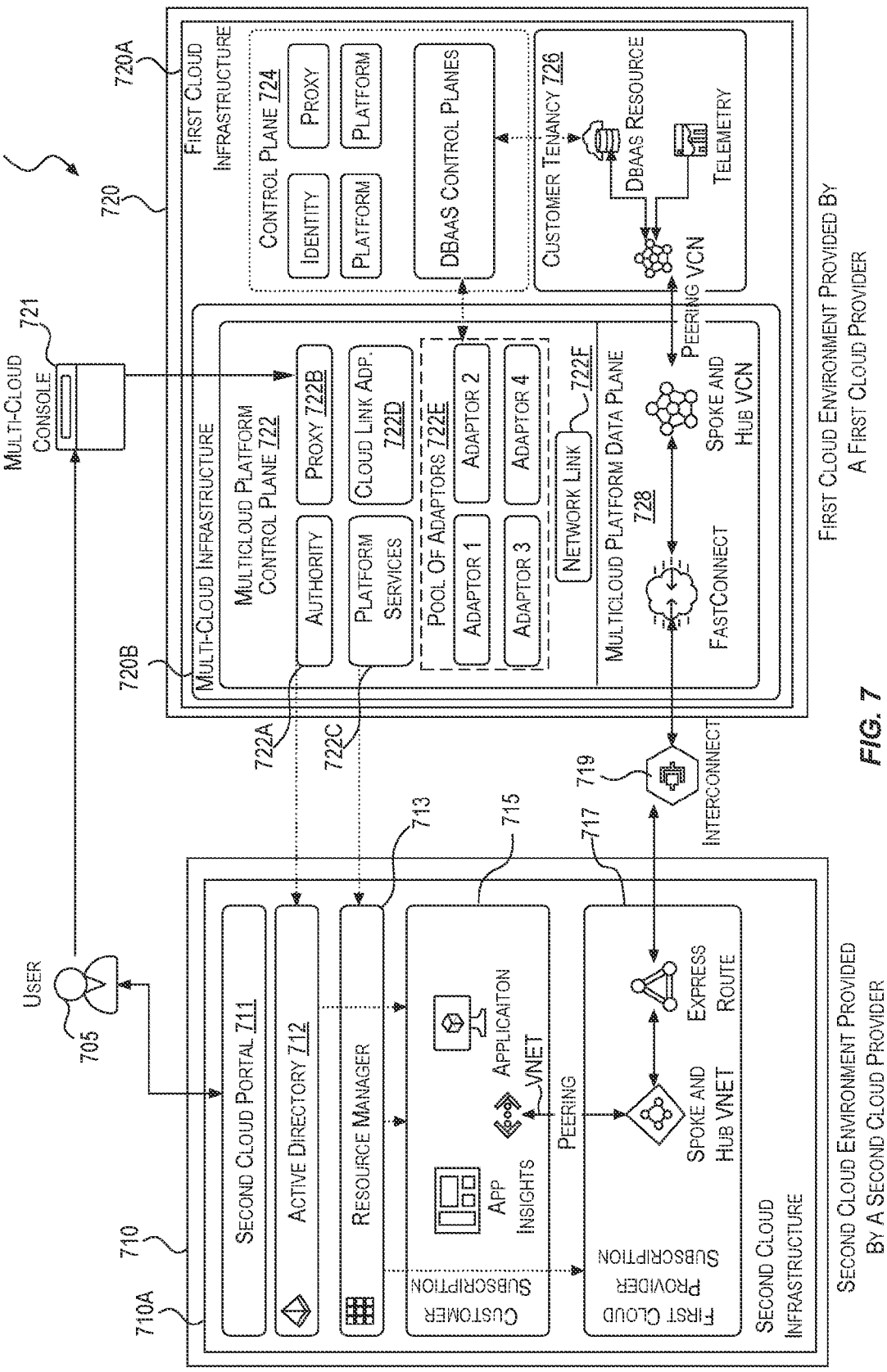
700
720A
720
FIRST CLOUD INFRASTRUCTURE
CONTROL PLANE 724
IDENTITY
PLATFORM
PROXY
PLATFORM
DBAAS CONTROL PLANES
CUSTOMER TENANCY 726
DBAAS RESOURCE
TELEMETRY
PEERING VCN
MULTI-CLOUD CONSOLE
721
MULTI-CLOUD INFRASTRUCTURE
MULTICLOUD PLATFORM CONTROL PLANE 722
AUTHORITY
PROXY 722B
PLATFORM SERVICES
CLOUD LINK ADP. 722D
POOL OF ADAPTORS 722E
ADAPTOR 1
ADAPTOR 2
ADAPTOR 3
ADAPTOR 4
NETWORK LINK 722F
MULTICLOUD PLATFORM DATA PLANE 728
FASTCONNECT
SPOKE AND HUB VCN
720B
722A
722C
719
INTERCONNECT
FIRST CLOUD ENVIRONMENT PROVIDED BY A FIRST CLOUD PROVIDER
713
715
717
USER
705
710
710A
SECOND CLOUD PORTAL 711
ACTIVE DIRECTORY 712
RESOURCE MANAGER
CUSTOMER SUBSCRIPTION
APP INSIGHTS
APPLICAITON VNET
PEERING
FIRST CLOUD PROVIDER SUBSCRIPTION
SPOKE AND HUB VNET
EXPRESS ROUTE
SECOND CLOUD INFRASTRUCTURE
SECOND CLOUD ENVIRONMENT PROVIDED BY A SECOND CLOUD PROVIDER

FIG. 7

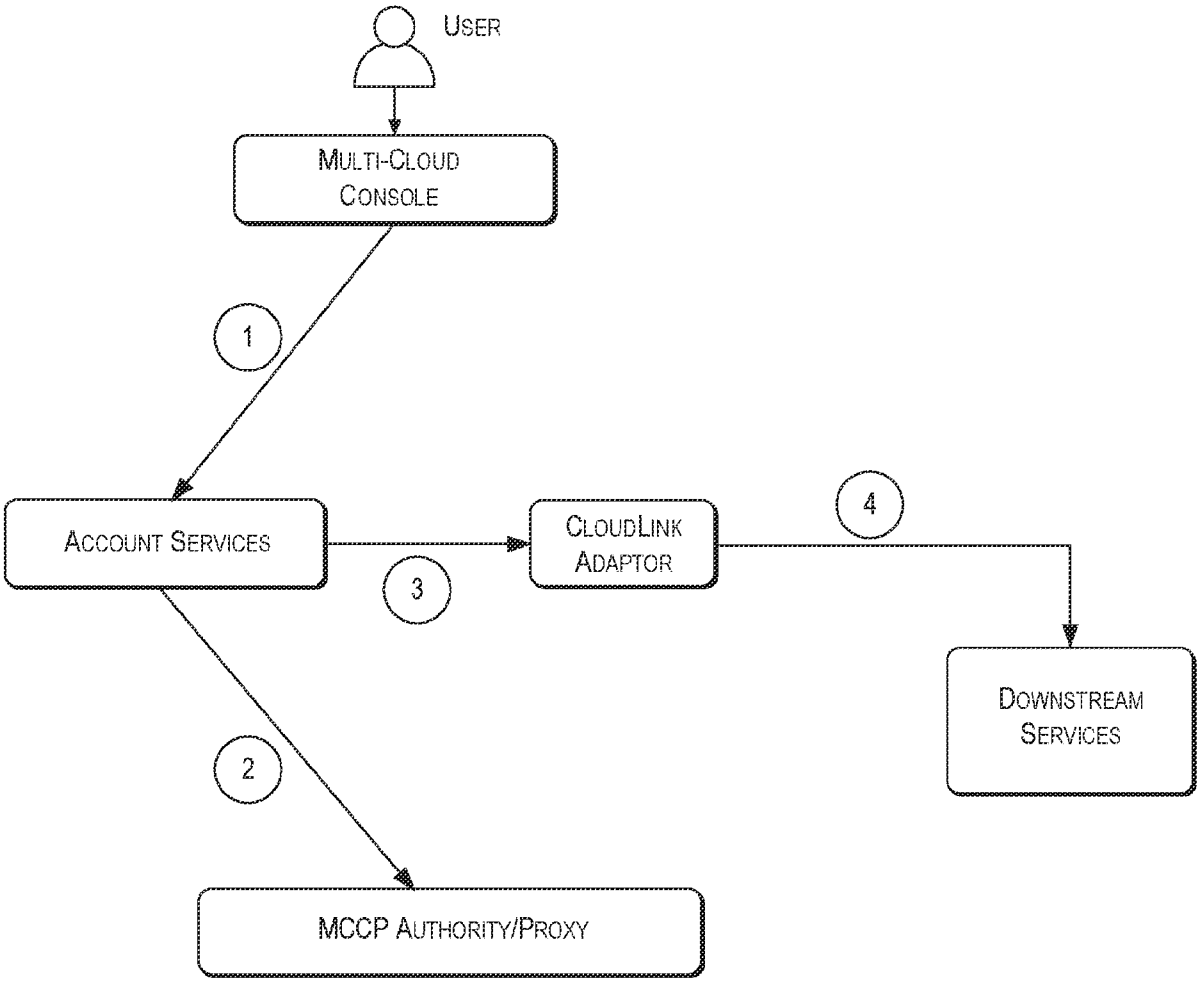
USER
MULTI-CLOUD CONSOLE
1
ACCOUNT SERVICES
3
CLOUDLINK ADAPTOR
4
DOWNSTREAM SERVICES
2
MCCP AUTHORITY/PROXY

*FIG. 8A*

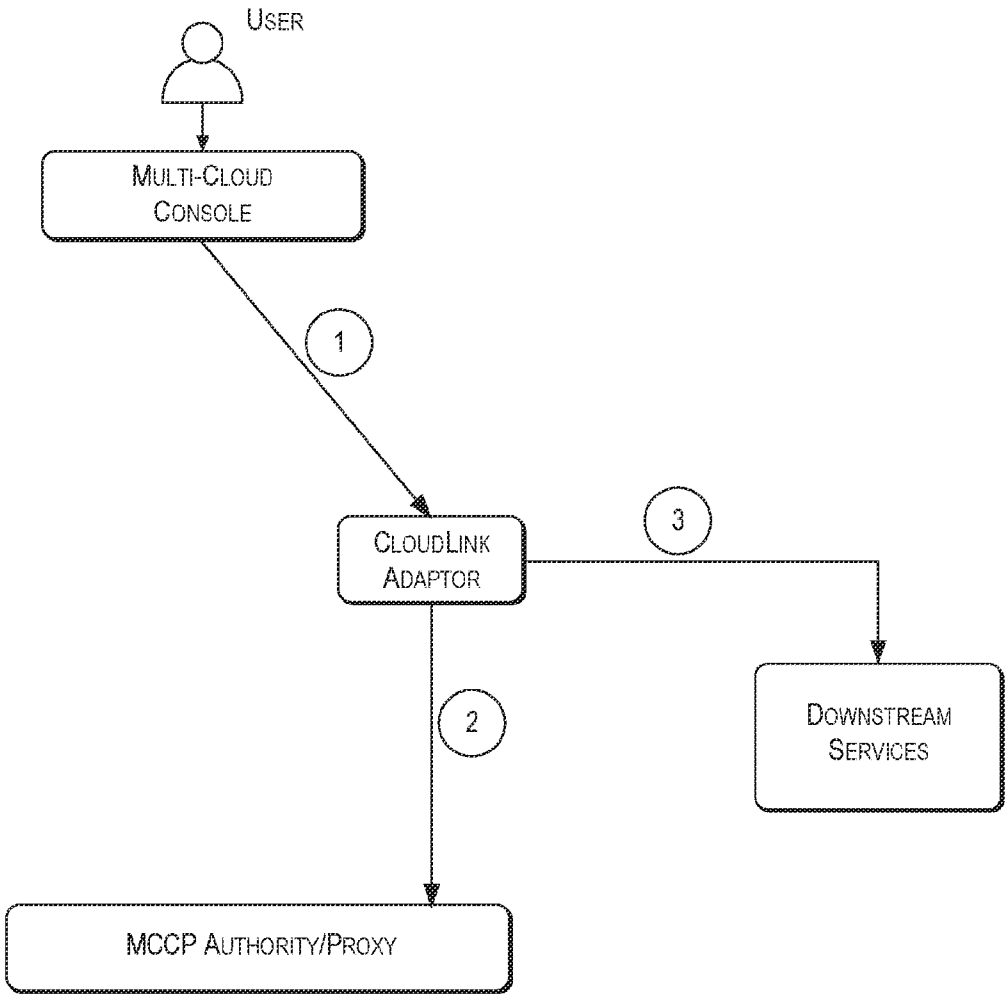
USER
MULTI-CLOUD CONSOLE
1
CLOUDLINK ADAPTOR
3
2
DOWNSTREAM SERVICES
MCCP AUTHORITY/PROXY

*FIG. 8B*

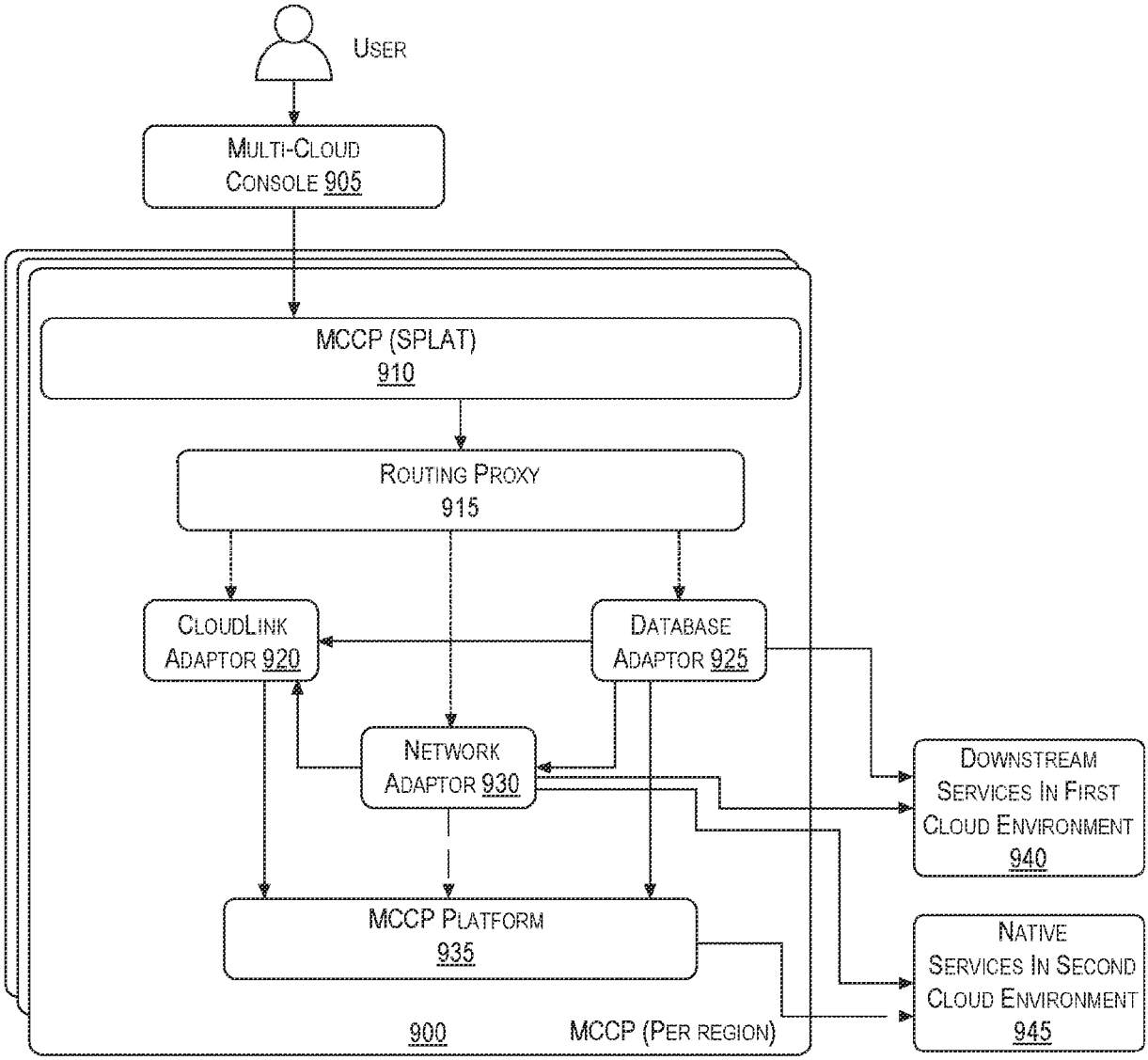
USER
MULTI-CLOUD CONSOLE 905
MCCP (SPLAT) 910
ROUTING PROXY 915
CLOUDLINK ADAPTOR 920
DATABASE ADAPTOR 925
NETWORK ADAPTOR 930
MCCP PLATFORM 935
900
MCCP (PER REGION)
DOWNSTREAM SERVICES IN FIRST CLOUD ENVIRONMENT 940
NATIVE SERVICES IN SECOND CLOUD ENVIRONMENT 945

*FIG. 9*

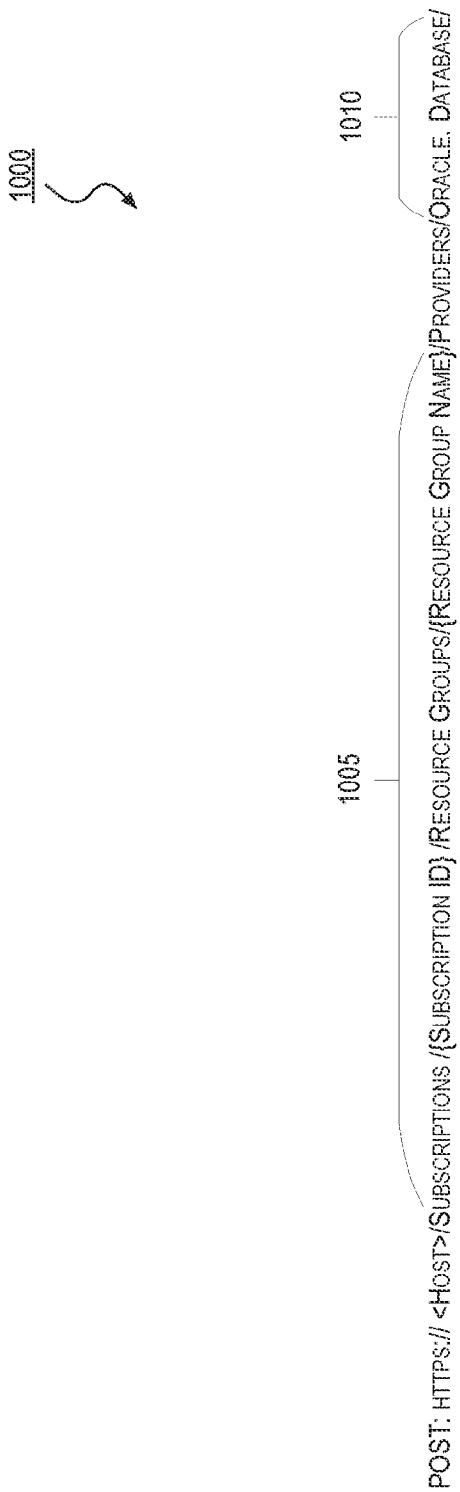
1000
1010
1005
POST: HTTPS:// <HOST>/SUBSCRIPTIONS /{SUBSCRIPTION ID} /RESOURCE GROUPS/{RESOURCE GROUP NAME}/PROVIDERS/ORACLE. DATABASE/
EXADATA INFRASTRUCTURE/{NAME} / VERSION = 2121-03-01
1015

FIG. 10

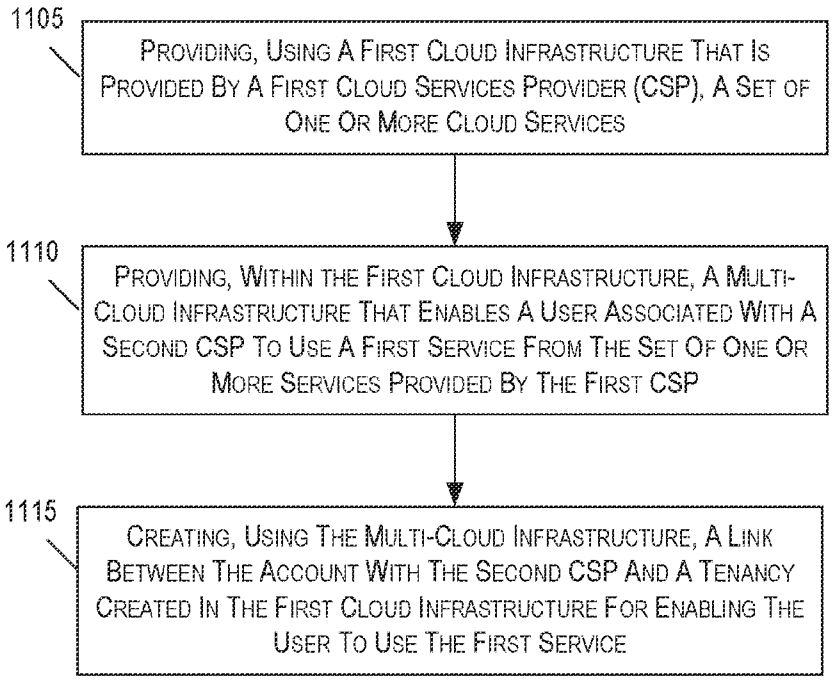
1105
PROVIDING, USING A FIRST CLOUD INFRASTRUCTURE THAT IS PROVIDED BY A FIRST CLOUD SERVICES PROVIDER (CSP), A SET OF ONE OR MORE CLOUD SERVICES
1110
PROVIDING, WITHIN THE FIRST CLOUD INFRASTRUCTURE, A MULTI-CLOUD INFRASTRUCTURE THAT ENABLES A USER ASSOCIATED WITH A SECOND CSP TO USE A FIRST SERVICE FROM THE SET OF ONE OR MORE SERVICES PROVIDED BY THE FIRST CSP
1115
CREATING, USING THE MULTI-CLOUD INFRASTRUCTURE, A LINK BETWEEN THE ACCOUNT WITH THE SECOND CSP AND A TENANCY CREATED IN THE FIRST CLOUD INFRASTRUCTURE FOR ENABLING THE USER TO USE THE FIRST SERVICE

*FIG. 11*

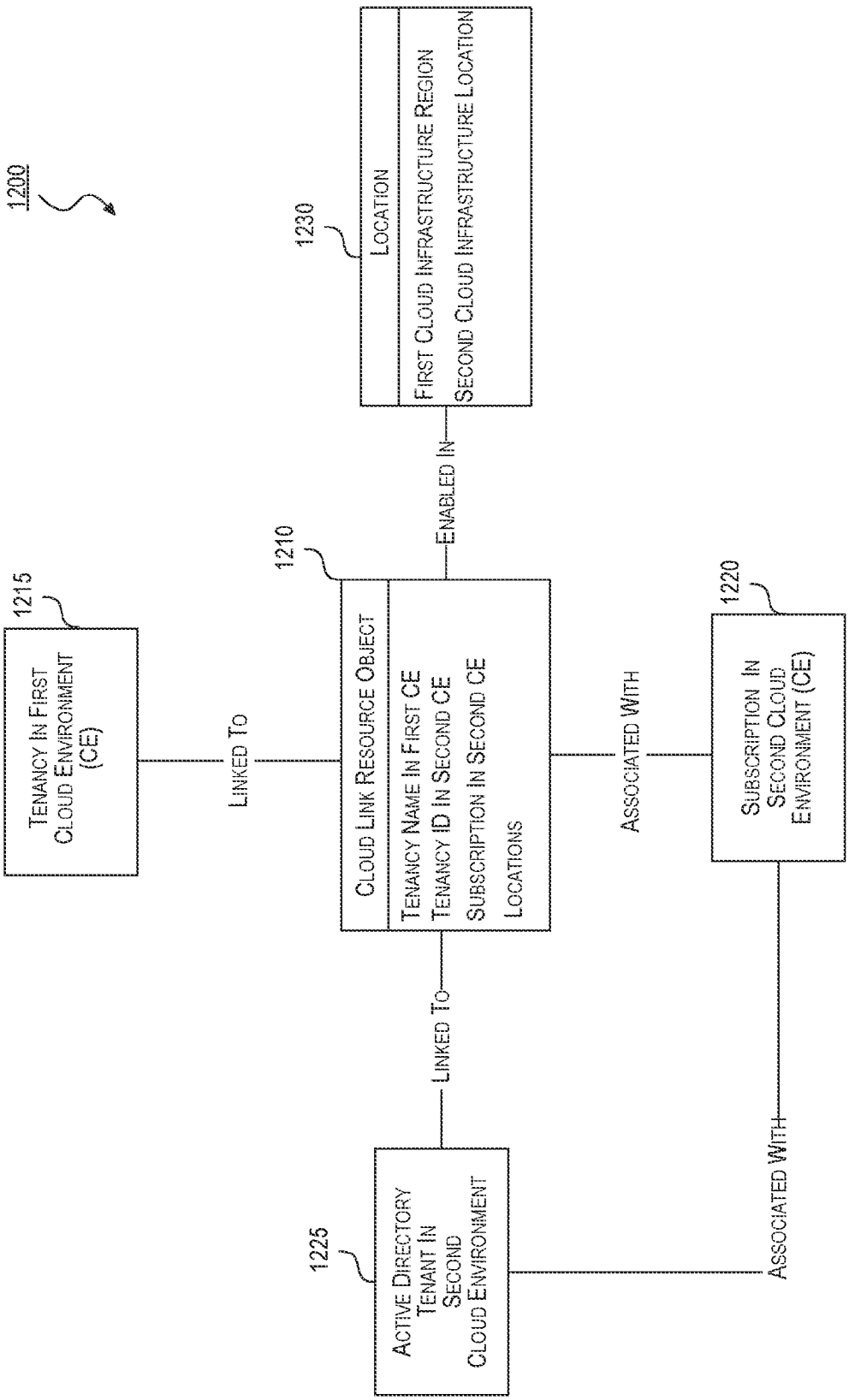
1200
1230
LOCATION
FIRST CLOUD INFRASTRUCTURE REGION
SECOND CLOUD INFRASTRUCTURE LOCATION
ENABLED IN
1210
CLOUD LINK RESOURCE OBJECT
TENANCY NAME IN FIRST CE
TENANCY ID IN SECOND CE
SUBSCRIPTION IN SECOND CE
LOCATIONS
1215
TENANCY IN FIRST CLOUD ENVIRONMENT (CE)
LINKED TO
ASSOCIATED WITH
1220
SUBSCRIPTION IN SECOND CLOUD ENVIRONMENT (CE)
LINKED TO
1225
ACTIVE DIRECTORY TENANT IN SECOND CLOUD ENVIRONMENT
ASSOCIATED WITH

FIG. 12

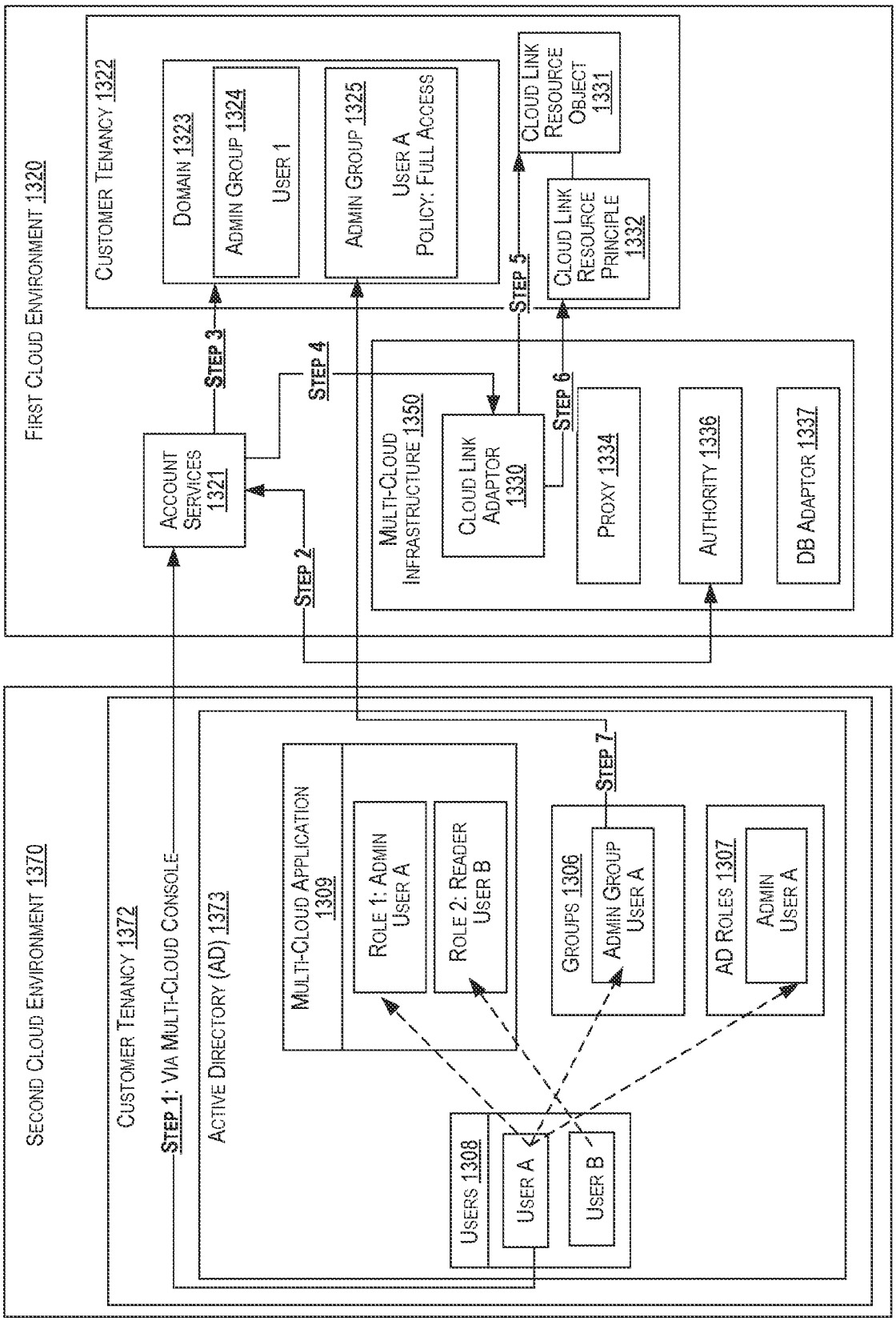
First Cloud Environment 1320
Customer Tenancy 1322
Domain 1323
Admin Group 1324
User 1
Admin Group 1325
User A
Policy: Full Access
Cloud Link Resource Object 1331
Cloud Link Resource Principle 1332
Step 3
Step 4
Step 5
Step 6
Account Services 1321
Step 2
Multi-Cloud Infrastructure 1350
Cloud Link Adaptor 1330
Proxy 1334
Authority 1336
DB Adaptor 1337
Second Cloud Environment 1370
Customer Tenancy 1372
Step 1: Via Multi-Cloud Console
Active Directory (AD) 1373
Multi-Cloud Application 1309
Role 1: Admin User A
Role 2: Reader User B
Step 7
Groups 1306
Admin Group User A
AD Roles 1307
Admin User A
Users 1308
User A
User B

FIG. 13

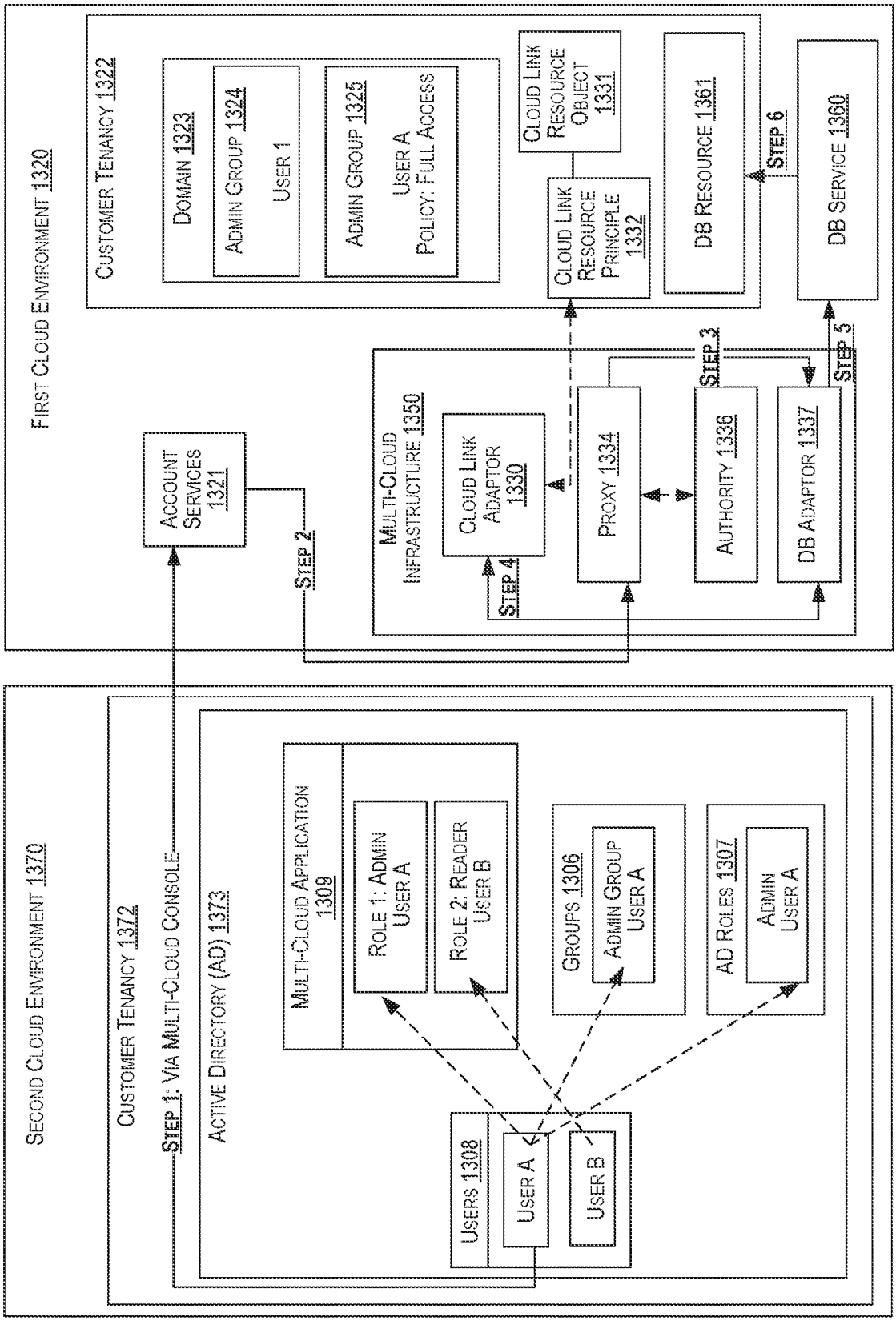
First Cloud Environment 1320
Customer Tenancy 1322
Domain 1323
Admin Group 1324
User 1
Admin Group 1325
User A
Policy: Full Access
Cloud Link Resource Object 1331
Cloud Link Resource Principle 1332
DB Resource 1361
DB Service 1360
Step 6
Account Services 1321
Step 2
Multi-Cloud Infrastructure 1350
Cloud Link Adaptor 1330
Step 4
Proxy 1334
Step 3
Authority 1336
DB Adaptor 1337
Step 5
Second Cloud Environment 1370
Customer Tenancy 1372
Step 1: Via Multi-Cloud Console
Active Directory (AD) 1373
Multi-Cloud Application 1309
Role 1: Admin User A
Role 2: Reader User B
Groups 1306
Admin Group User A
AD Roles 1307
Admin User A
Users 1308
User A
User B

FIG. 14

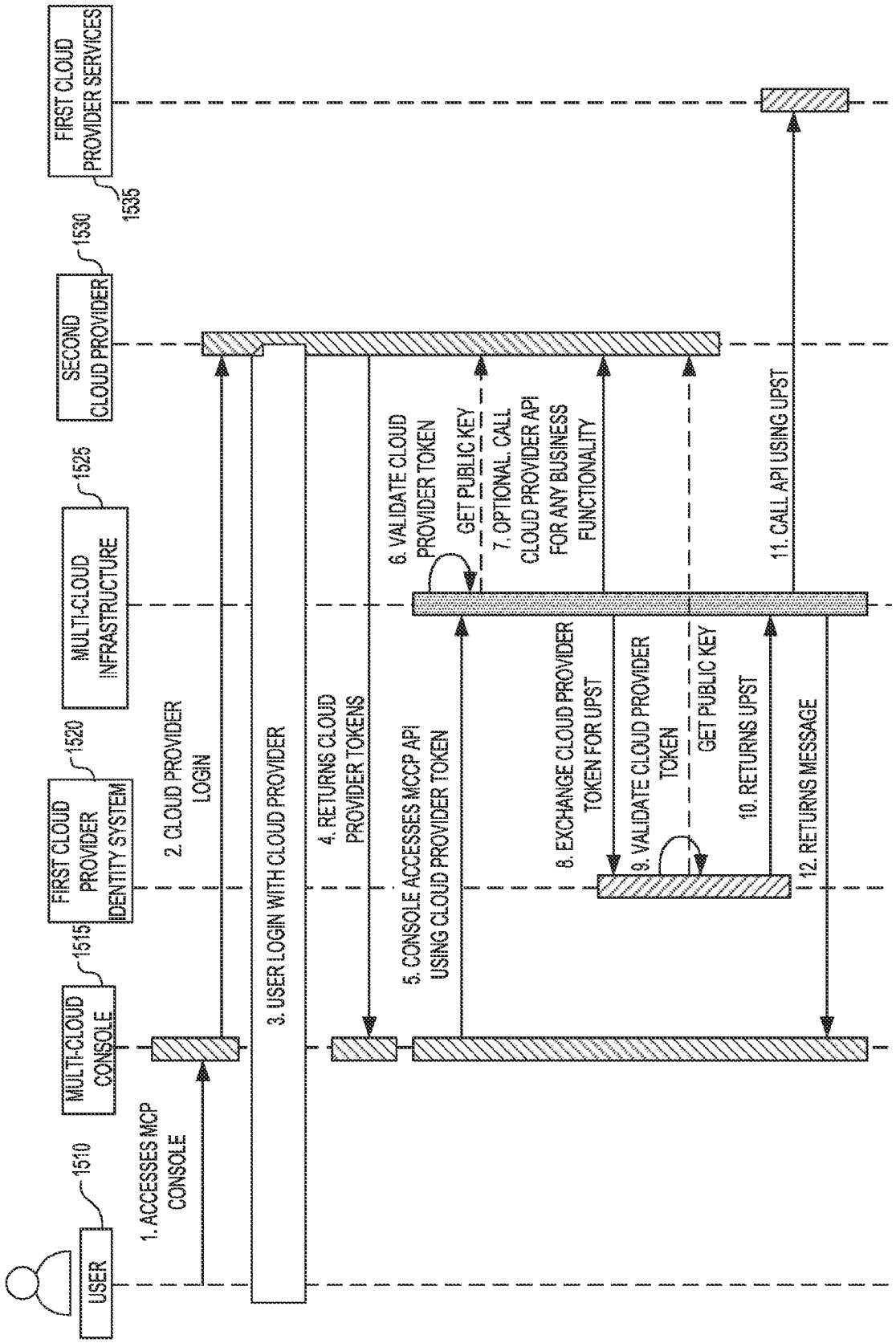
USER
1510
MULTI-CLOUD CONSOLE
1515
FIRST CLOUD PROVIDER IDENTITY SYSTEM
1520
MULTI-CLOUD INFRASTRUCTURE
1525
SECOND CLOUD PROVIDER
1530
FIRST CLOUD PROVIDER SERVICES
1535
1. ACCESSES MCP CONSOLE
2. CLOUD PROVIDER LOGIN
3. USER LOGIN WITH CLOUD PROVIDER
4. RETURNS CLOUD PROVIDER TOKENS
5. CONSOLE ACCESSES MCCP API USING CLOUD PROVIDER TOKEN
6. VALIDATE CLOUD PROVIDER TOKEN
GET PUBLIC KEY
7. OPTIONAL CALL CLOUD PROVIDER API FOR ANY BUSINESS FUNCTIONALITY
8. EXCHANGE CLOUD PROVIDER TOKEN FOR UPST
9. VALIDATE CLOUD PROVIDER TOKEN
GET PUBLIC KEY
10. RETURNS UPST
11. CALL API USING UPST
12. RETURNS MESSAGE

FIG. 15

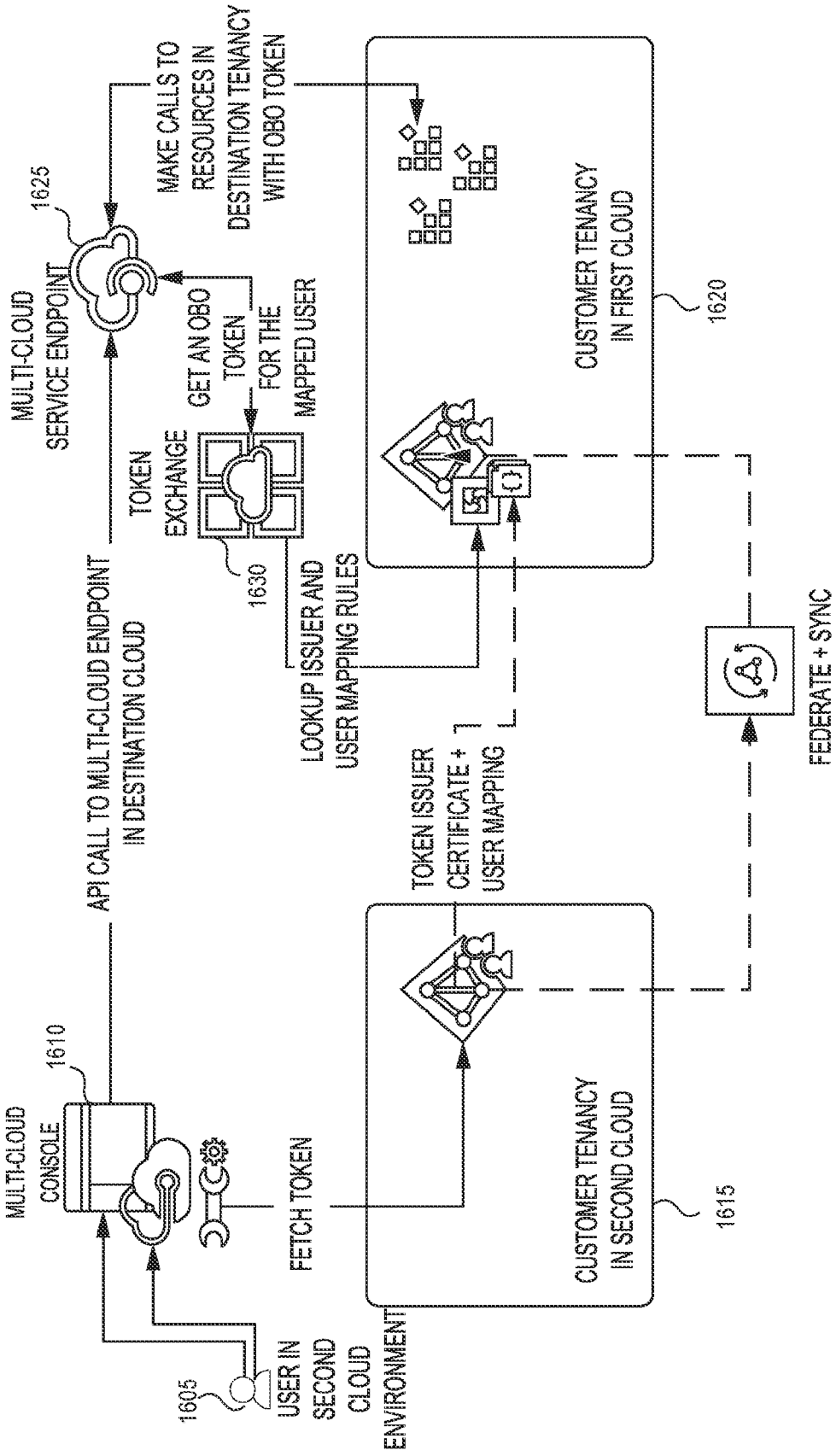
USER IN SECOND CLOUD ENVIRONMENT
1605
MULTI-CLOUD CONSOLE
1610
FETCH TOKEN
API CALL TO MULTI-CLOUD ENDPOINT IN DESTINATION CLOUD
MULTI-CLOUD SERVICE ENDPOINT
1625
TOKEN EXCHANGE
1630
GET AN OBO TOKEN FOR THE MAPPED USER
MAKE CALLS TO RESOURCES IN DESTINATION TENANCY WITH OBO TOKEN
LOOKUP ISSUER AND USER MAPPING RULES
TOKEN ISSUER CERTIFICATE + USER MAPPING
CUSTOMER TENANCY IN SECOND CLOUD
1615
CUSTOMER TENANCY IN FIRST CLOUD
1620
FEDERATE + SYNC

FIG. 16

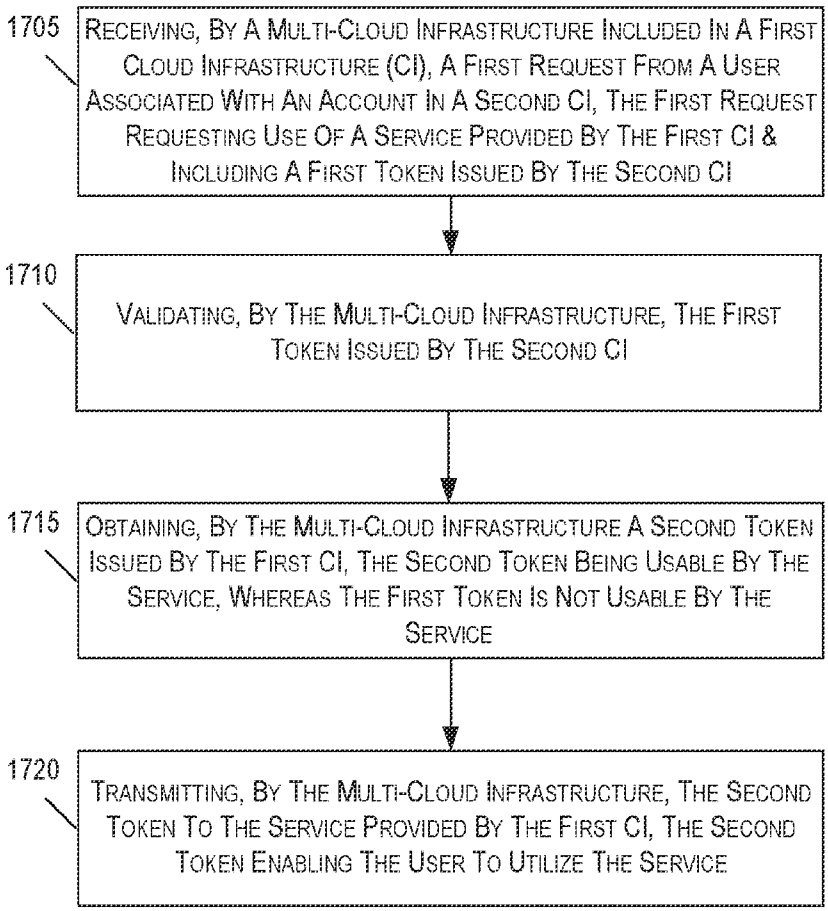
1705
RECEIVING, BY A MULTI-CLOUD INFRASTRUCTURE INCLUDED IN A FIRST CLOUD INFRASTRUCTURE (CI), A FIRST REQUEST FROM A USER ASSOCIATED WITH AN ACCOUNT IN A SECOND CI, THE FIRST REQUEST REQUESTING USE OF A SERVICE PROVIDED BY THE FIRST CI & INCLUDING A FIRST TOKEN ISSUED BY THE SECOND CI
1710
VALIDATING, BY THE MULTI-CLOUD INFRASTRUCTURE, THE FIRST TOKEN ISSUED BY THE SECOND CI
1715
OBTAINING, BY THE MULTI-CLOUD INFRASTRUCTURE A SECOND TOKEN ISSUED BY THE FIRST CI, THE SECOND TOKEN BEING USABLE BY THE SERVICE, WHEREAS THE FIRST TOKEN IS NOT USABLE BY THE SERVICE
1720
TRANSMITTING, BY THE MULTI-CLOUD INFRASTRUCTURE, THE SECOND TOKEN TO THE SERVICE PROVIDED BY THE FIRST CI, THE SECOND TOKEN ENABLING THE USER TO UTILIZE THE SERVICE

*FIG. 17*

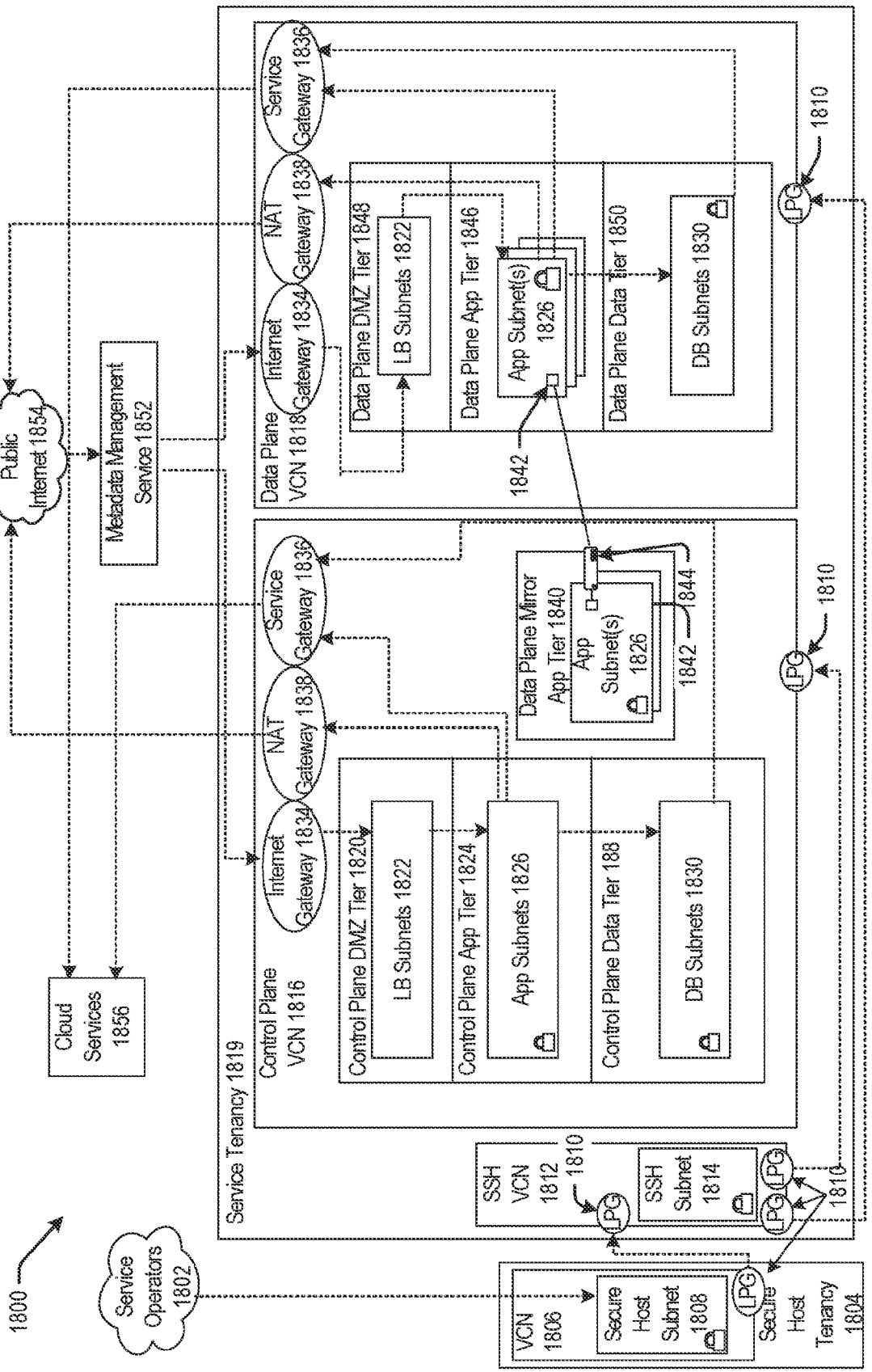
1800
Service Operators 1802
Public Internet 1854
Metadata Management Service 1852
Cloud Services 1856
Service Tenancy 1819
Control Plane VCN 1816
Control Plane DMZ Tier 1820
LB Subnets 1822
Internet Gateway 1834
NAT Gateway 1838
Service Gateway 1836
Control Plane App Tier 1824
App Subnets 1826
Control Plane Data Tier 188
DB Subnets 1830
Data Plane Mirror App Tier 1840
App Subnet(s) 1826
1842
1844
Data Plane VCN 1818
Data Plane DMZ Tier 1848
Data Plane App Tier 1846
Data Plane Data Tier 1850
1810
SSH VCN 1812
SSH Subnet 1814
VCN 1806
Secure Host Subnet 1808
Secure Host Tenancy 1804
LPG
PG

FIG. 18

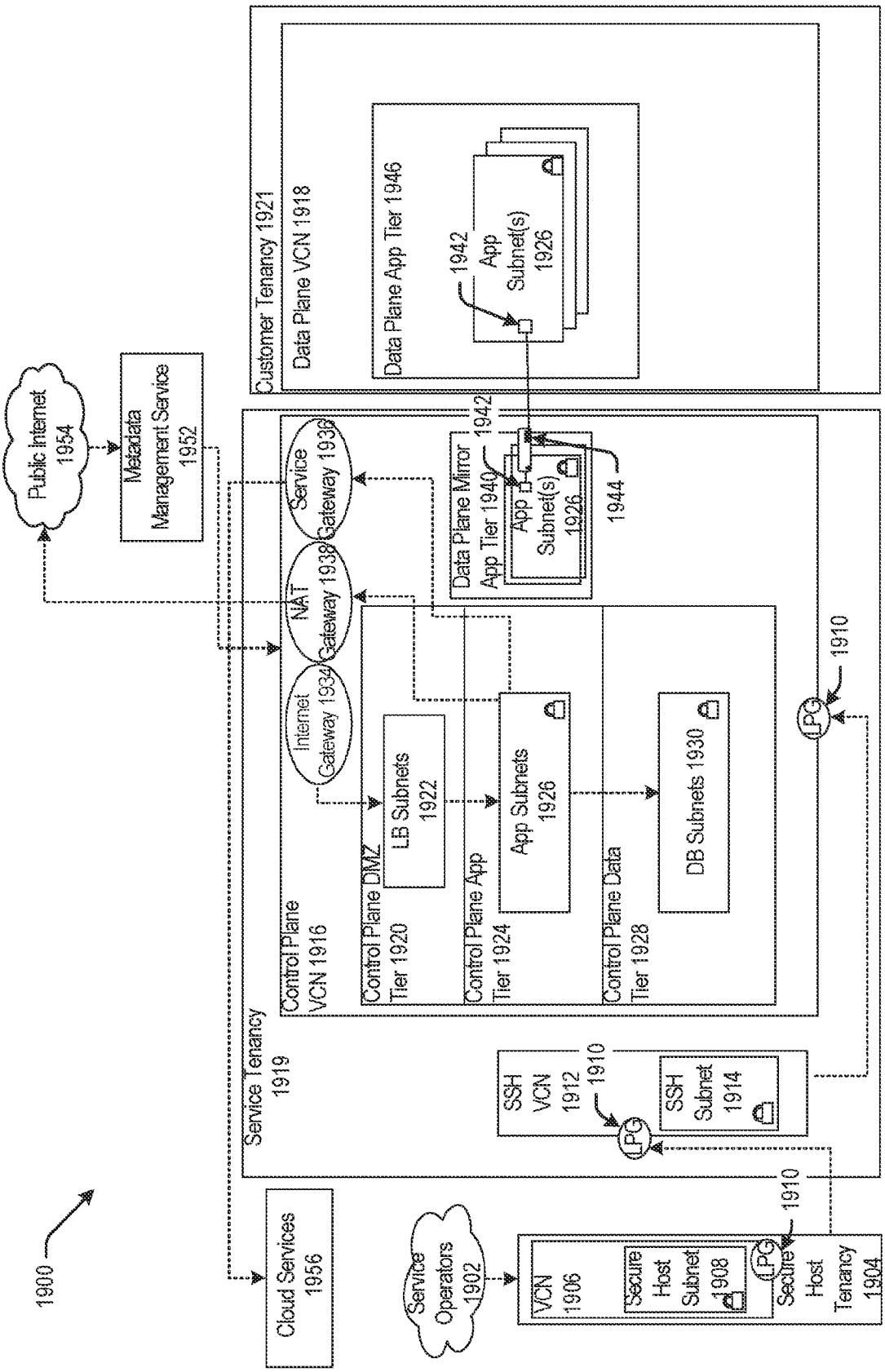
1900
Public Internet 1954
Metadata Management Service 1952
Customer Tenancy 1921
Data Plane VCN 1918
Data Plane App Tier 1946
1942
App Subnet(s) 1926
Service Gateway 1938
NAT Gateway 1936
Internet Gateway 1934
Data Plane Mirror App Tier 1940
1942
App Subnet(s) 1926
1944
Service Tenancy 1919
Control Plane VCN 1916
Control Plane DMZ Tier 1920
LB Subnets 1922
Control Plane App Tier 1924
App Subnets 1926
Control Plane Data Tier 1928
DB Subnets 1930
1910
SSH VCN 1912
1910
SSH Subnet 1914
1910
Cloud Services 1956
Service Operators 1902
VCN 1906
Secure Host Subnet 1908
Secure Host Tenancy 1904

FIG. 19

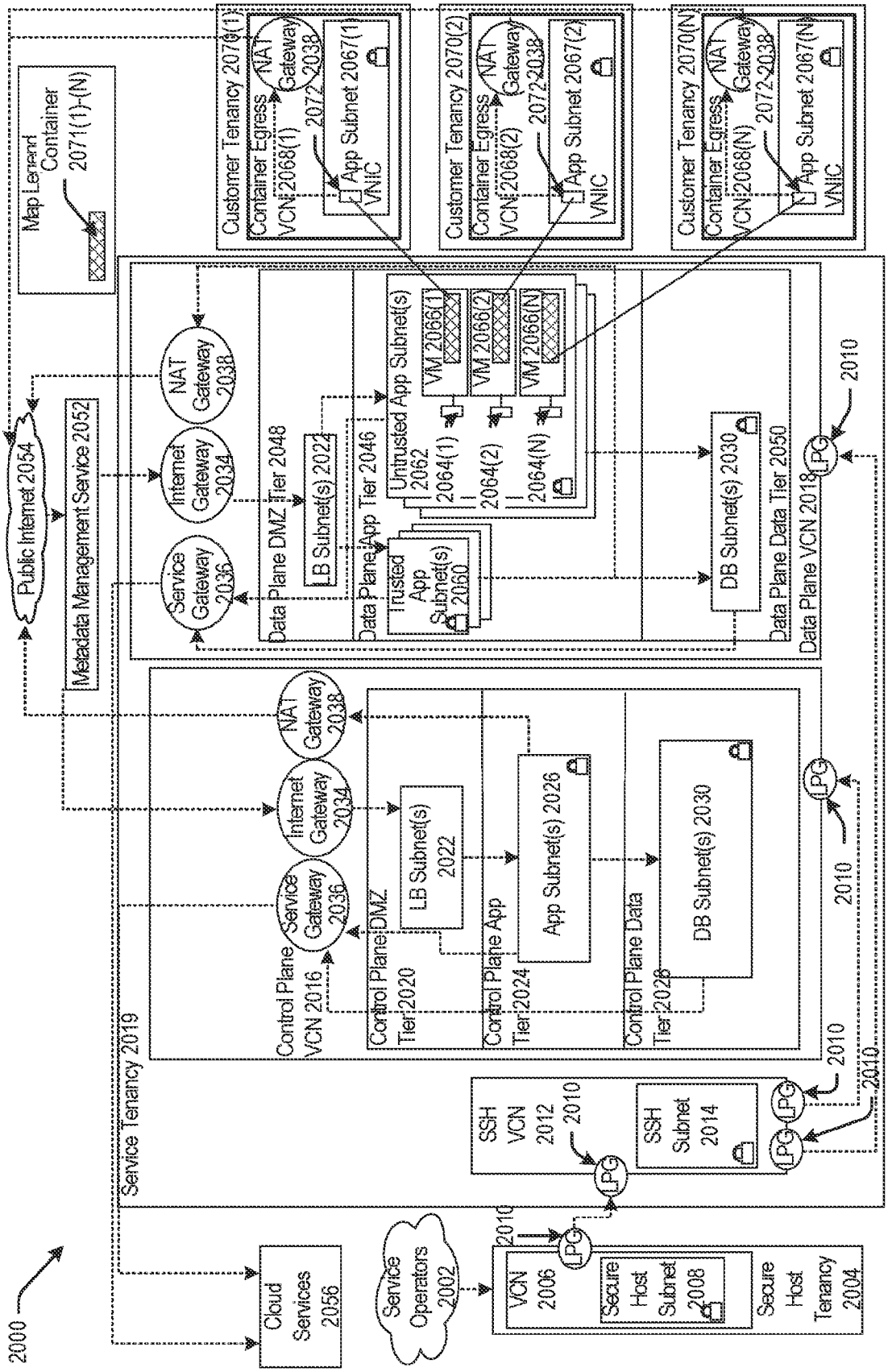
2000
Service Operators 2002
Cloud Services 2056
Service Tenancy 2019
Public Internet 2054
Metadata Management Service 2052
Service Gateway 2036
Internet Gateway 2034
NAT Gateway 2038
Map Legend
Container 2071(1)-(N)
Customer Tenancy 2070(1)
Container Egress VCN 2068(1)
2072
App Subnet 2067(1) VNIC
Customer Tenancy 2070(2)
Container Egress VCN 2068(2)
App Subnet 2067(2) VNIC
Customer Tenancy 2070(N)
Container Egress VCN 2068(N)
App Subnet 2067(N) VNIC
Data Plane DMZ Tier 2048
LB Subnet(s) 2022
Data Plane App Tier 2046
Untrusted App Subnet(s) 2062
Trusted App Subnet(s) 2060
VM 2066(1)
VM 2066(2)
VM 2066(N)
2064(1)
2064(2)
2064(N)
Data Plane Data Tier 2050
DB Subnet(s) 2030
Data Plane VCN 2018
2010
Control Plane VCN 2016
Control Plane DMZ Tier 2020
LB Subnet(s) 2022
Control Plane App Tier 2024
App Subnet(s) 2026
Control Plane Data Tier 2028
DB Subnet(s) 2030
SSH VCN 2012
SSH Subnet 2014
VCN 2006
Secure Host Subnet 2008
Secure Host Tenancy 2004
PG

FIG. 20

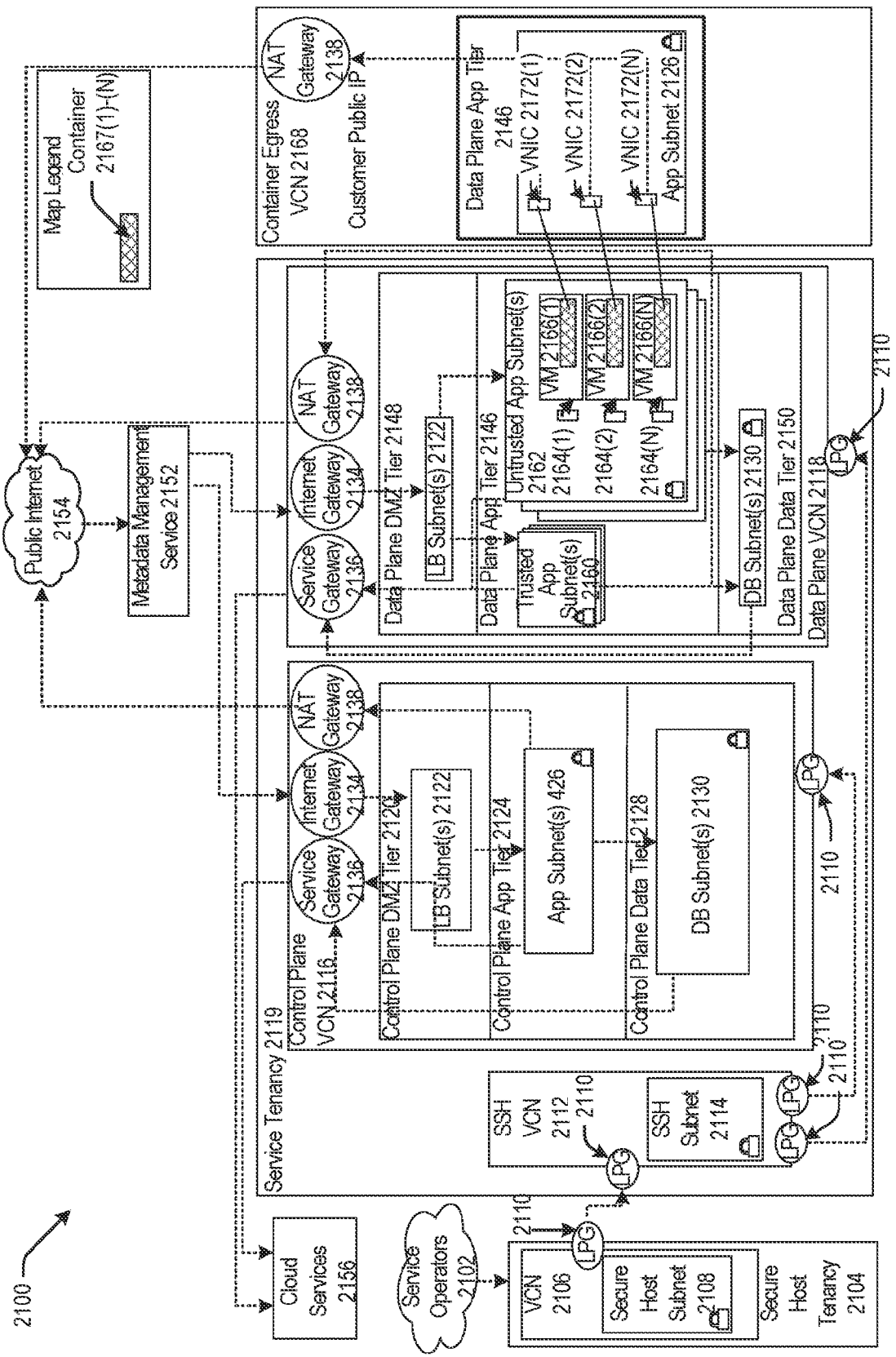
2100
Map Legend
Container
2167(1)-(N)
Container Egress
VCN 2168
NAT
Gateway
2138
Customer Public IP
Data Plane App Tier
2146
VNIC 2172(1)
VNIC 2172(2)
VNIC 2172(N)
App Subnet 2126
Public Internet
2154
Metadata Management
Service 2152
Service
Gateway
2136
Internet
Gateway
2134
NAT
Gateway
2138
Data Plane DMZ Tier 2148
LB Subnet(s) 2122
Data Plane App Tier 2146
Untrusted App Subnet(s)
2162
VM 2166(1)
VM 2166(2)
VM 2166(N)
2164(1)
2164(2)
2164(N)
Trusted
App
Subnet(s)
2160
DB Subnet(s) 2130
Data Plane Data Tier 2150
Data Plane VCN 2118
2110
LPG
Service Tenancy 2119
Control Plane
VCN 2116
Control Plane DMZ Tier 2120
LB Subnet(s) 2122
Control Plane App Tier 2124
App Subnet(s) 426
Control Plane Data Tier 2128
DB Subnet(s) 2130
SSH
VCN
2112
2110
SSH
Subnet
2114
PG
Cloud
Services
2156
Service
Operators
2102
VCN
2106
Secure
Host
Subnet
2108
Secure
Host
Tenancy
2104

FIG. 21

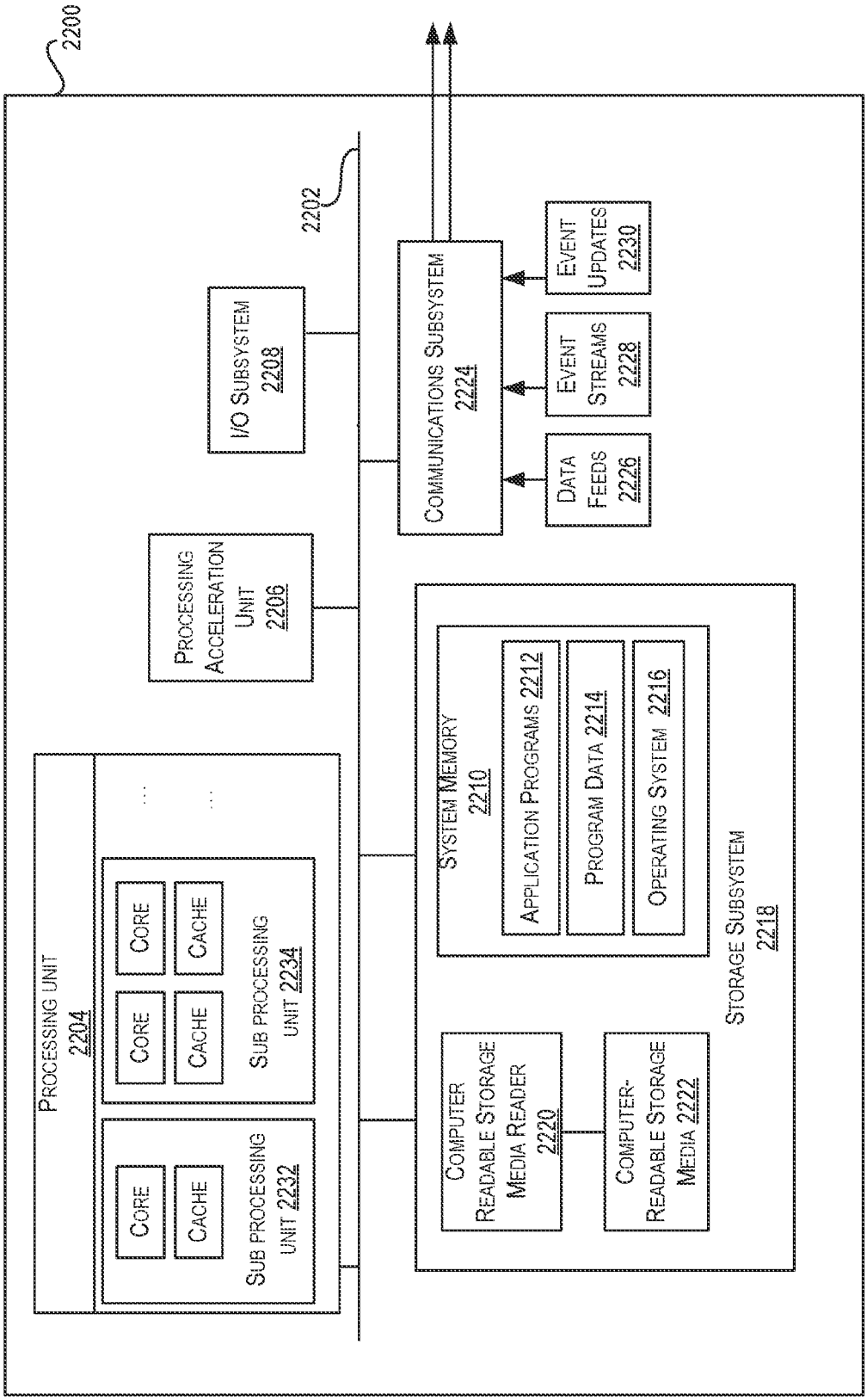
2200
2202
I/O SUBSYSTEM 2208
COMMUNICATIONS SUBSYSTEM 2224
EVENT UPDATES 2230
EVENT STREAMS 2228
DATA FEEDS 2226
PROCESSING ACCELERATION UNIT 2206
PROCESSING UNIT 2204
CORE
CACHE
SUB PROCESSING UNIT 2232
SUB PROCESSING UNIT 2234
SYSTEM MEMORY 2210
APPLICATION PROGRAMS 2212
PROGRAM DATA 2214
OPERATING SYSTEM 2216
STORAGE SUBSYSTEM 2218
COMPUTER READABLE STORAGE MEDIA READER 2220
COMPUTER-READABLE STORAGE MEDIA 2222

FIG. 22

# INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/061723

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/50 G06F21/30
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | SITARAM DINKAR ET AL: "Keystone federated security", 8TH INTERNATIONAL CONFERENCE FOR INTERNET TECHNOLOGY AND SECURED TRANSACTIONS (ICITST-2013), INFONOMICS SOCIETY, 9 December 2013 (2013-12-09), pages 659-664, XP032572354, DOI: 10.1109/ICITST.2013.6750285 [retrieved on 2014-02-27] the whole document ----- | 1-20 |

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 3 May 2023 | 12/05/2023 |

| Name and mailing address of the ISA/ | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040,<br>Fax: (+31-70) 340-3016 | Touloupis, Emmanuel |