US 20220385488A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2022/0385488 A1**

**Kothuri et al.** (43) **Pub. Date: Dec. 1, 2022**

(54) **SYSTEM AND METHOD FOR RECONCILING CONSUMPTION DATA**

(71) Applicant: **Nutanix, Inc.**, San Jose, CA (US)

(72) Inventors: **Venkata Vamsi Krishna Kothuri**, San Jose, CA (US); **Shi Shu**, Santa Clara, CA (US); **Manoj Badola**, Bangalore (IN); **Sravan Kumar Muthyala**, Bangalore (IN)

(73) Assignee: **Nutanix, Inc.**, San Jose, CA (US)

(21) Appl. No.: **17/375,910**

(22) Filed: **Jul. 14, 2021**

(30) **Foreign Application Priority Data**

May 31, 2021 (IN) ..... 202141024135

**Publication Classification**

(51) **Int. Cl.**
***H04L 12/14*** (2006.01)

(52) **U.S. Cl.**
CPC ..... ***H04L 12/14*** (2013.01)

(57) **ABSTRACT**

Various embodiments disclosed herein are related to a non-transitory computer readable storage medium. In some embodiments, the medium includes instructions stored thereon that, when executed by a processor, cause the processor to receive, at a server, from a first cluster of nodes on an edge network in communication with the server, first resource consumption data of a first service hosted on the edge network, calculate a first resource consumption quantity based on the first resource consumption data, receive, from a second cluster of nodes on the edge network, delayed resource consumption data of a second service hosted on the edge network, and calculate a second resource consumption quantity based on the delayed resource consumption data. In some embodiments, the first resource consumption data is collected at a first time and the delayed resource consumption data collected at the first time.

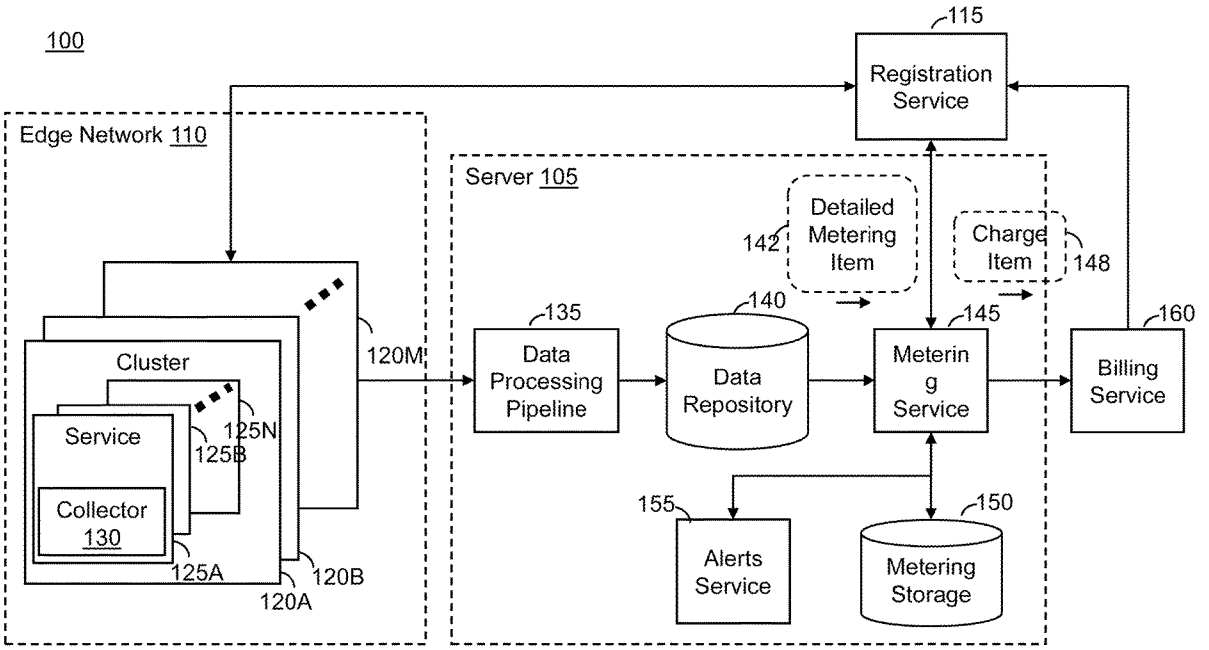


FIG. 1

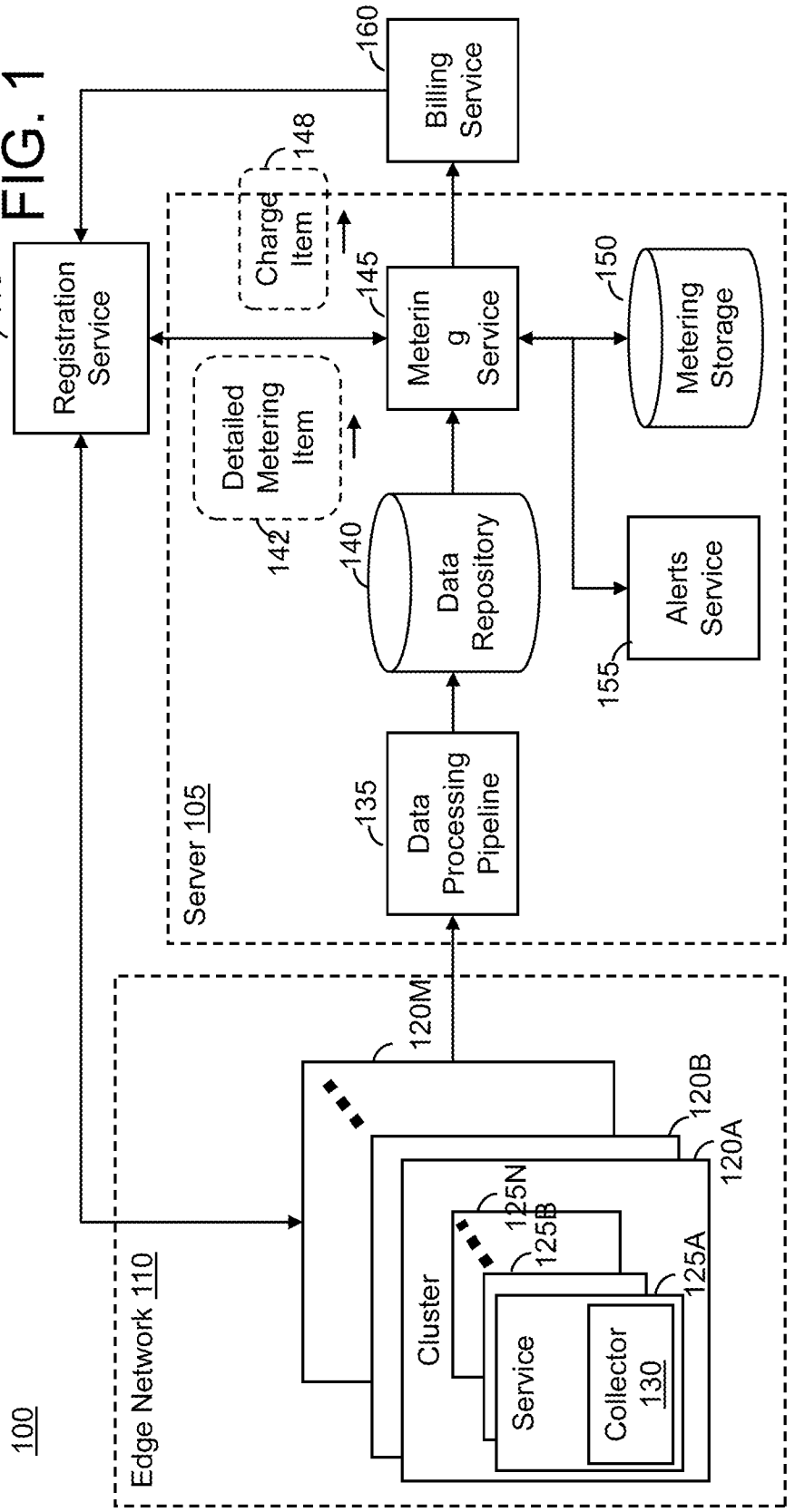

100
Edge Network 110
Cluster
Service
Collector 130
125A
125B
125N
120A
120B
120M
Server 105
Data Processing Pipeline
135
Data Repository
140
Detailed Metering Item
142
Charge Item
148
Meterin g Service
145
Metering Storage
150
Alerts Service
155
Billing Service
160
Registration Service
115

FIG. 2A

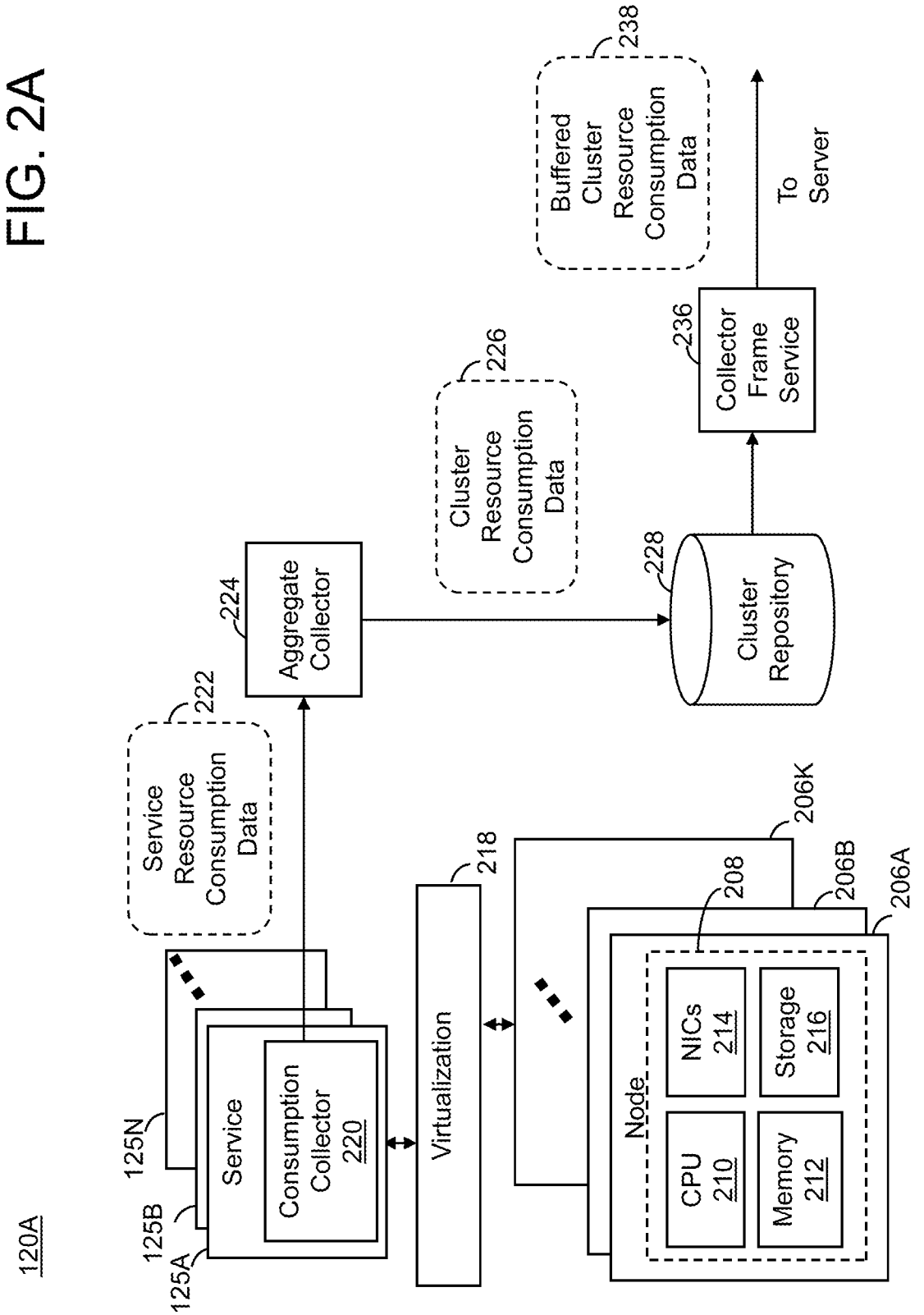

120A
125A
125B
125N
Service
Consumption Collector 220
Virtualization
218
206A
206B
206K
Node
CPU 210
Memory 212
NICs 214
208
Storage 216
Service Resource Consumption Data
222
Aggregate Collector
224
Cluster Resource Consumption Data
226
Cluster Repository
228
Collector Frame Service
236
Buffered Cluster Resource Consumption Data
238
To Server

FIG. 2B

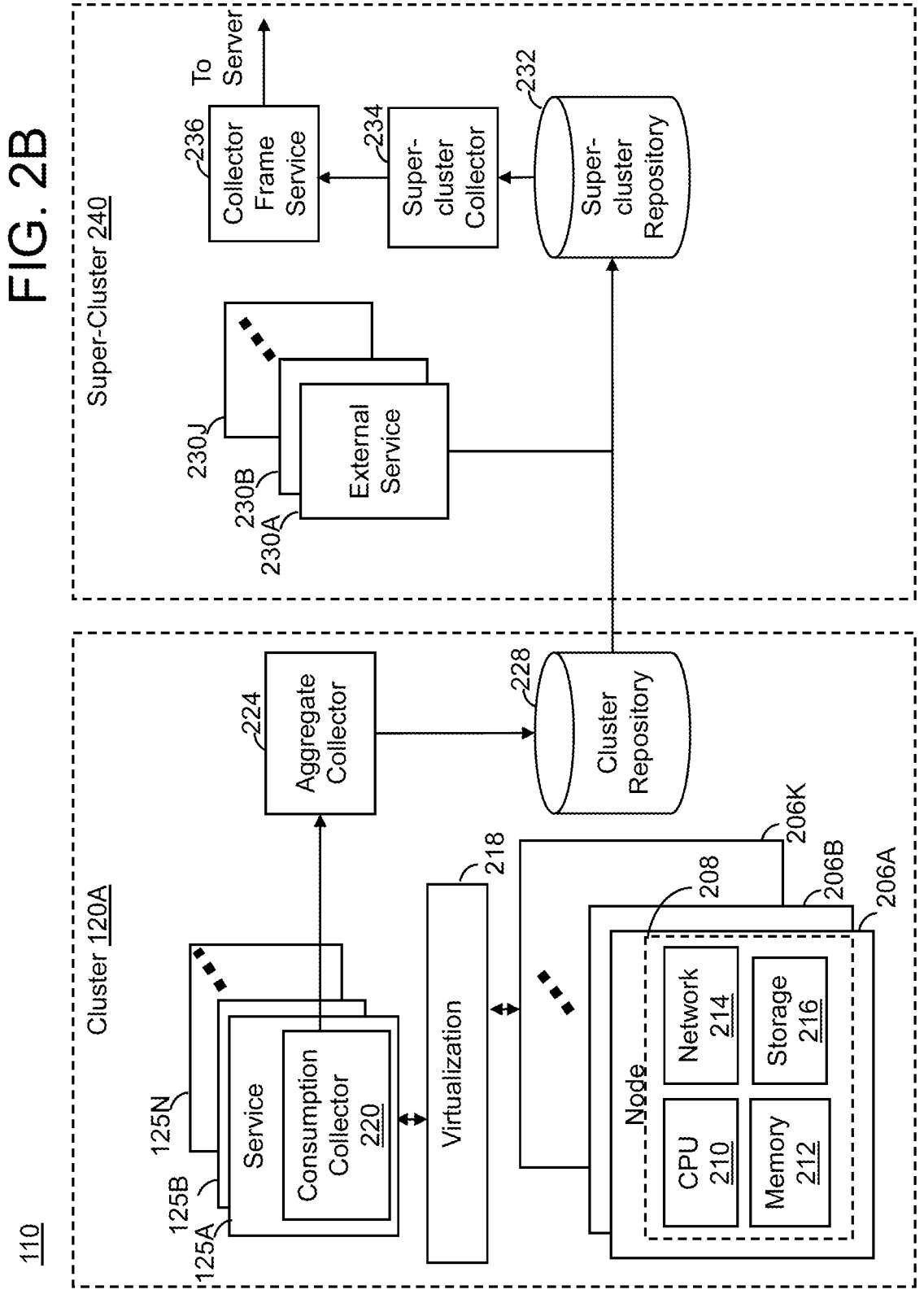

110
Cluster 120A
125A
125B
125N
Service
Consumption Collector 220
Virtualization
218
Node
CPU 210
Memory 212
Network 214
Storage 216
208
206A
206B
206K
Aggregate Collector
224
Cluster Repository
228
Super-Cluster 240
230A
230B
230J
External Service
Super-cluster Repository
232
Super-cluster Collector
234
Collector Frame Service
236
To Server

FIG. 3A

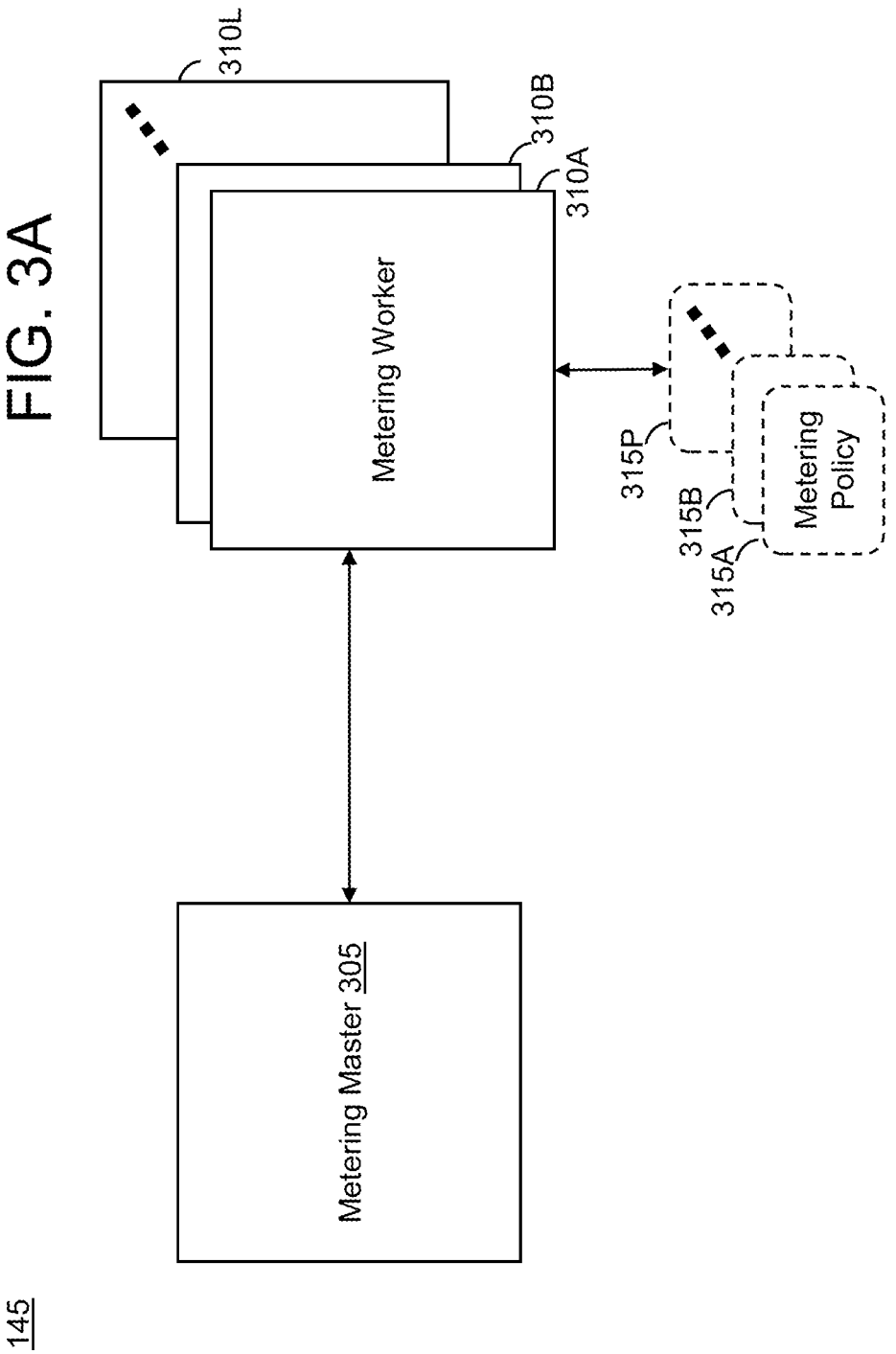

310L
310B
310A
Metering Worker
315P
315B
315A
Metering Policy
Metering Master 305
145

FIG. 3B

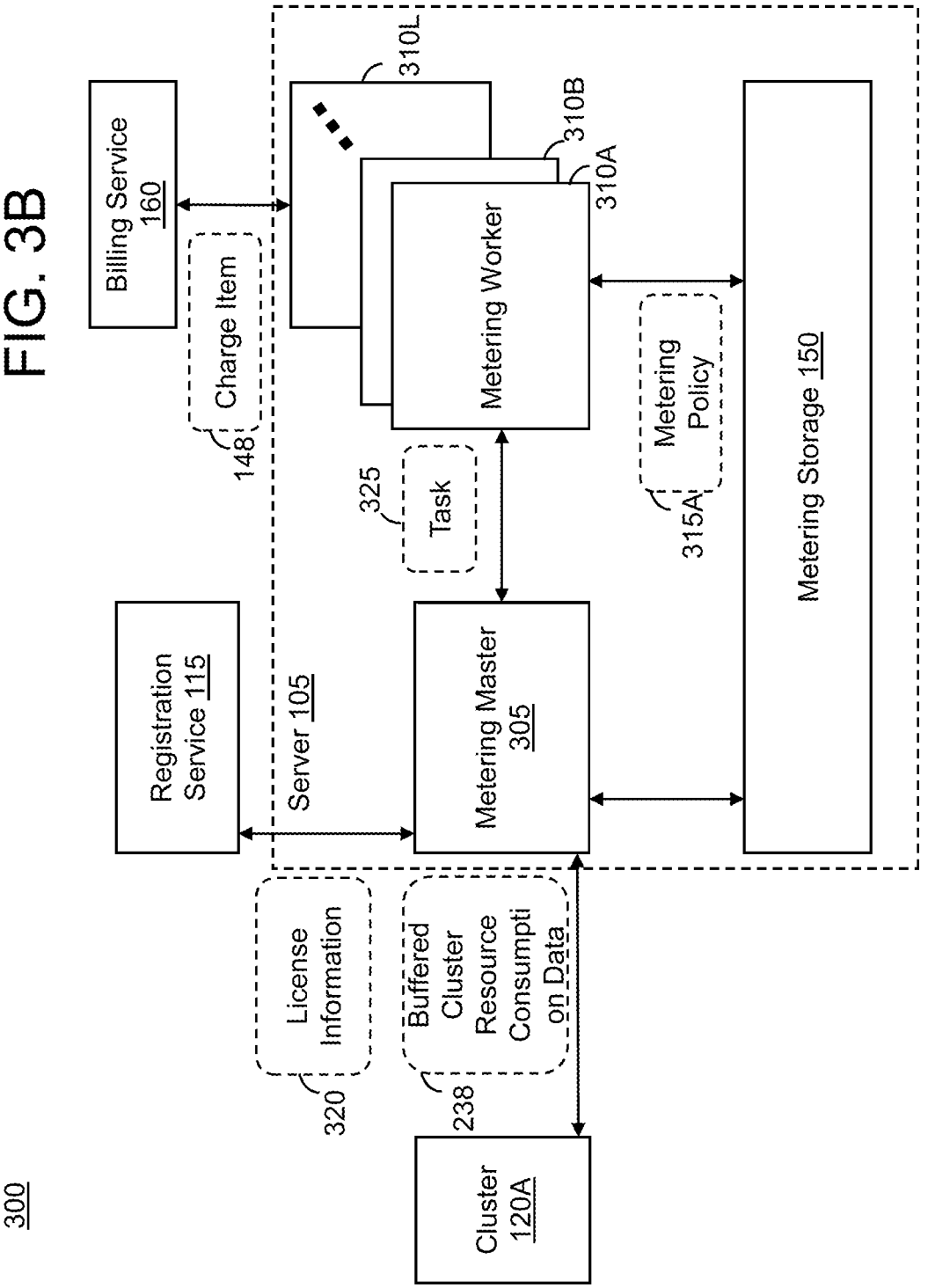
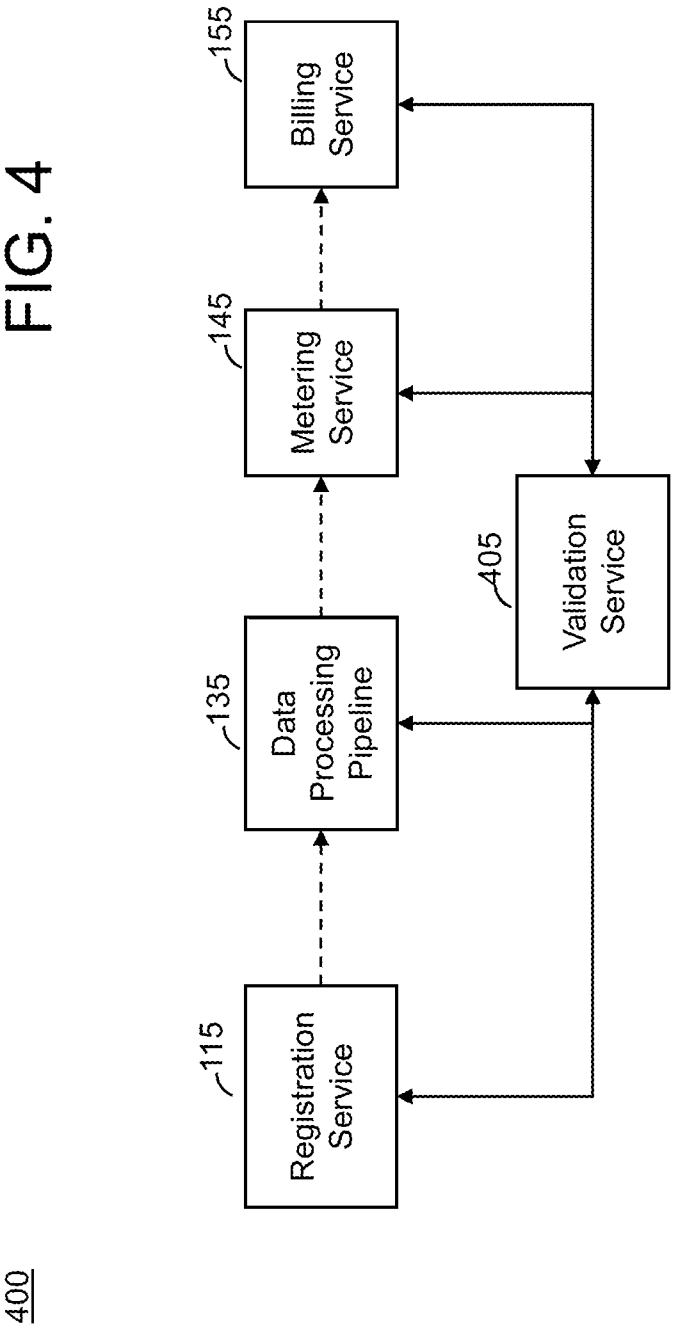

300
Billing Service 160
Charge Item
148
Server 105
310L
310B
310A
Metering Worker
325
Task
Metering Policy
315A
Metering Storage 150
Registration Service 115
Metering Master 305
License Information
320
Buffered Cluster Resource Consumption Data
238
Cluster 120A

FIG. 4


400
115
Registration Service
135
Data Processing Pipeline
145
Metering Service
155
Billing Service
405
Validation Service

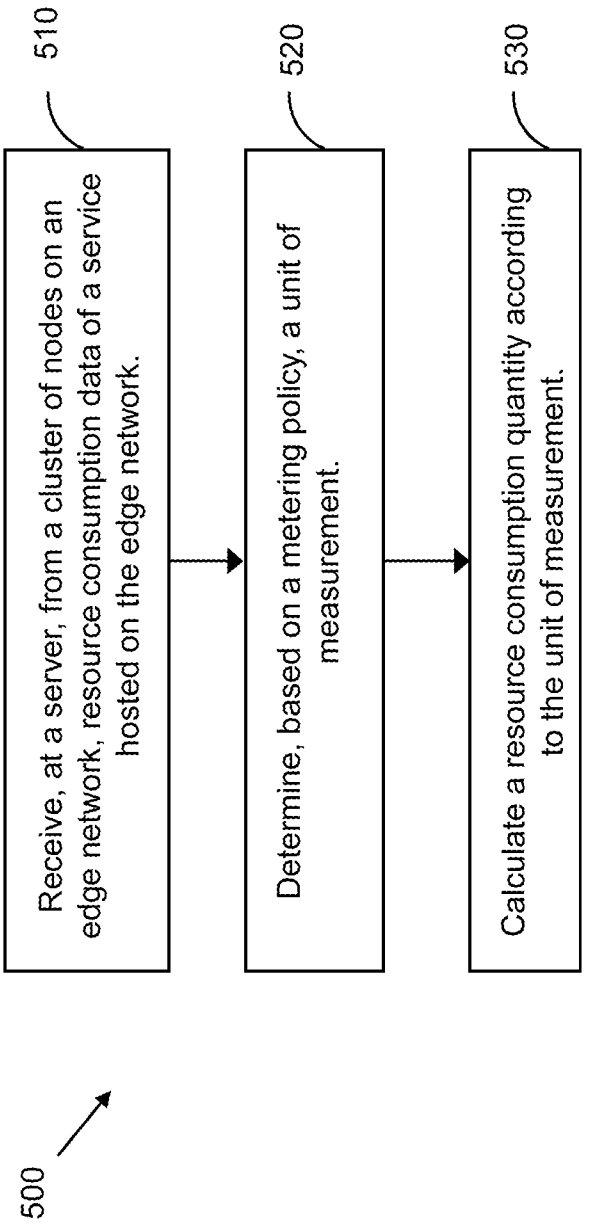

500
510
Receive, at a server, from a cluster of nodes on an edge network, resource consumption data of a service hosted on the edge network.
520
Determine, based on a metering policy, a unit of measurement.
530
Calculate a resource consumption quantity according to the unit of measurement.

FIG. 5

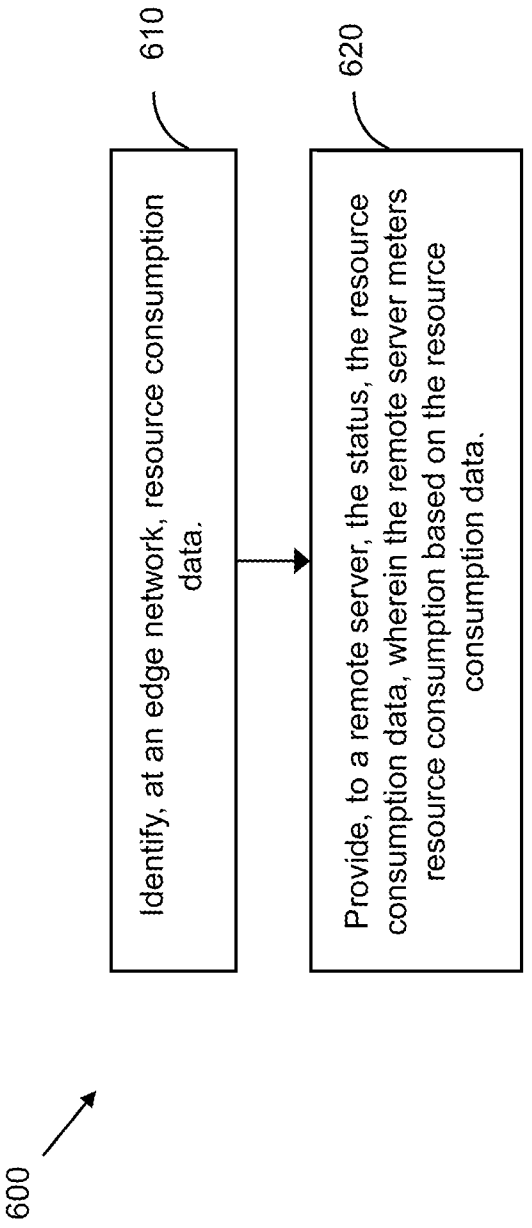

600
610
Identify, at an edge network, resource consumption data.
620
Provide, to a remote server, the status, the resource consumption data, wherein the remote server meters resource consumption based on the resource consumption data.

FIG. 6

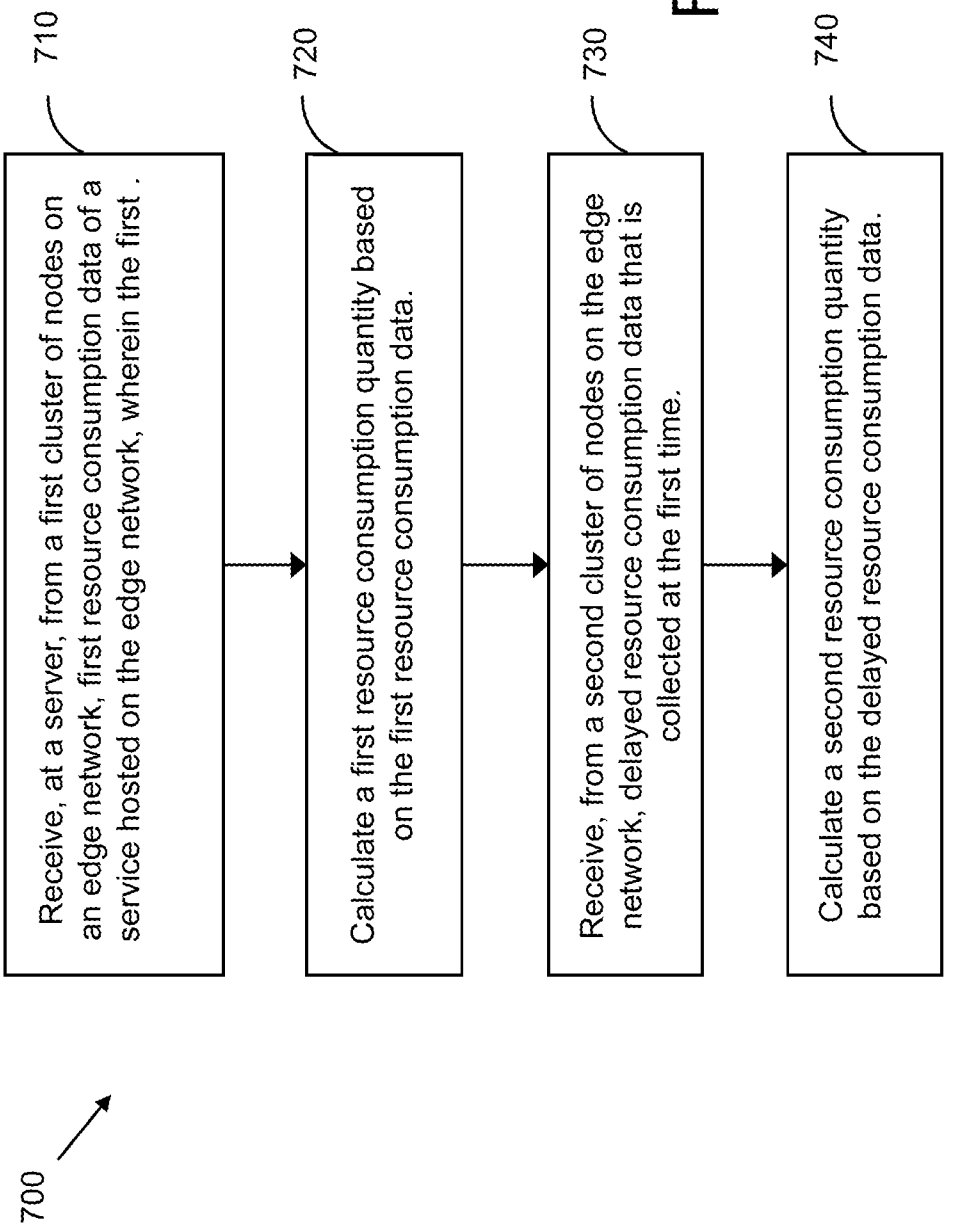

700
710
Receive, at a server, from a first cluster of nodes on an edge network, first resource consumption data of a service hosted on the edge network, wherein the first .
720
Calculate a first resource consumption quantity based on the first resource consumption data.
730
Receive, from a second cluster of nodes on the edge network, delayed resource consumption data that is collected at the first time.
740
Calculate a second resource consumption quantity based on the delayed resource consumption data.

FIG. 7

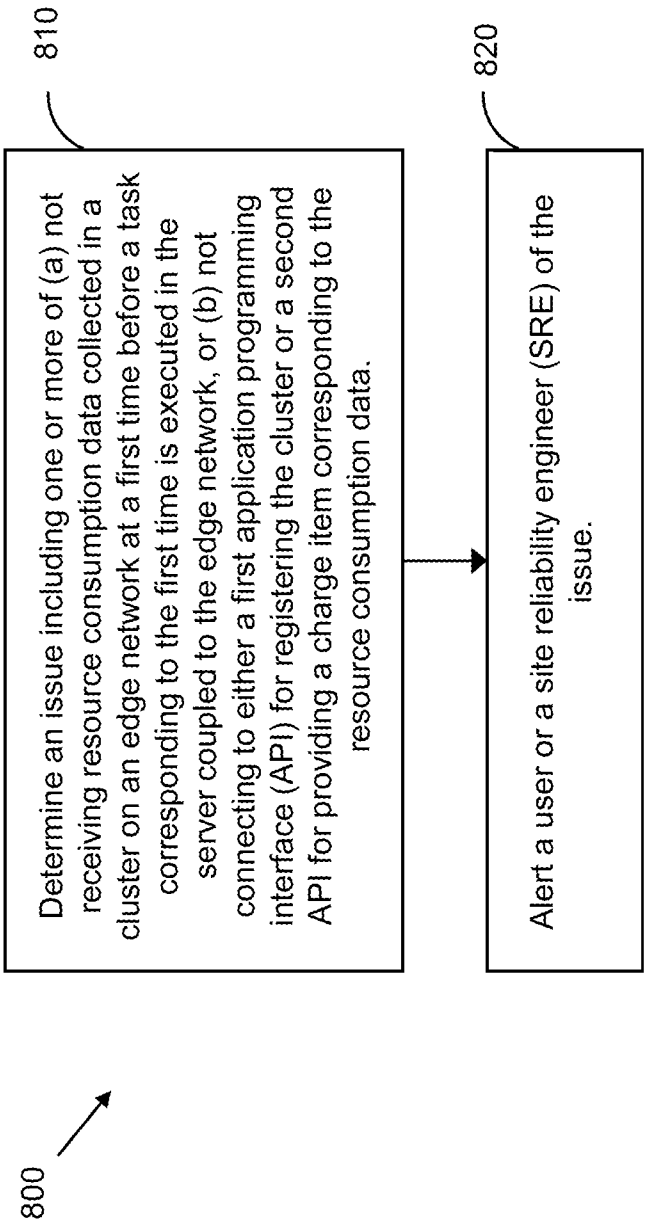

800
810
Determine an issue including one or more of (a) not receiving resource consumption data collected in a cluster on an edge network at a first time before a task corresponding to the first time is executed in the server coupled to the edge network, or (b) not connecting to either a first application programming interface (API) for registering the cluster or a second API for providing a charge item corresponding to the resource consumption data.
820
Alert a user or a site reliability engineer (SRE) of the issue.

FIG. 8

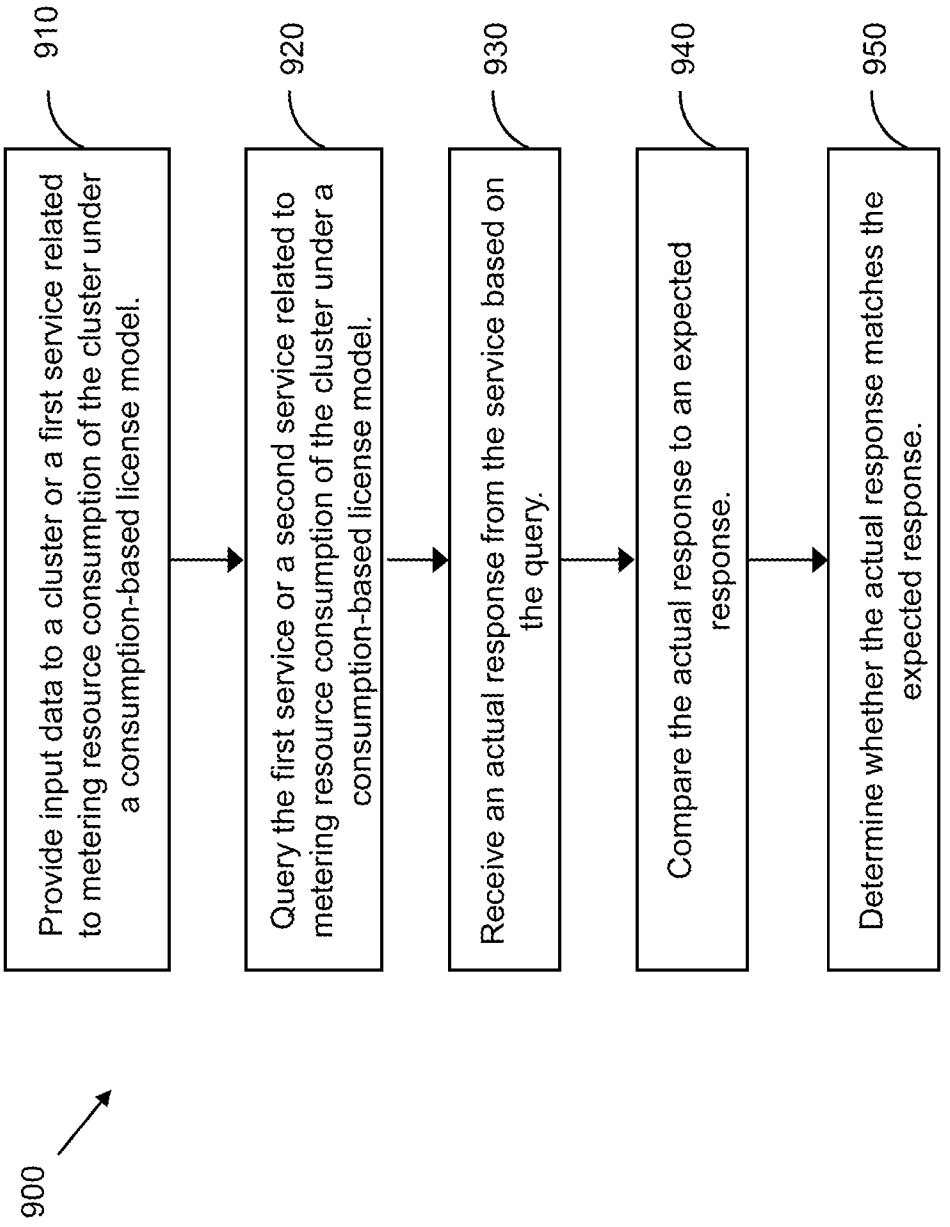

900
910
Provide input data to a cluster or a first service related to metering resource consumption of the cluster under a consumption-based license model.
920
Query the first service or a second service related to metering resource consumption of the cluster under a consumption-based license model.
930
Receive an actual response from the service based on the query.
940
Compare the actual response to an expected response.
950
Determine whether the actual response matches the expected response.

FIG. 9

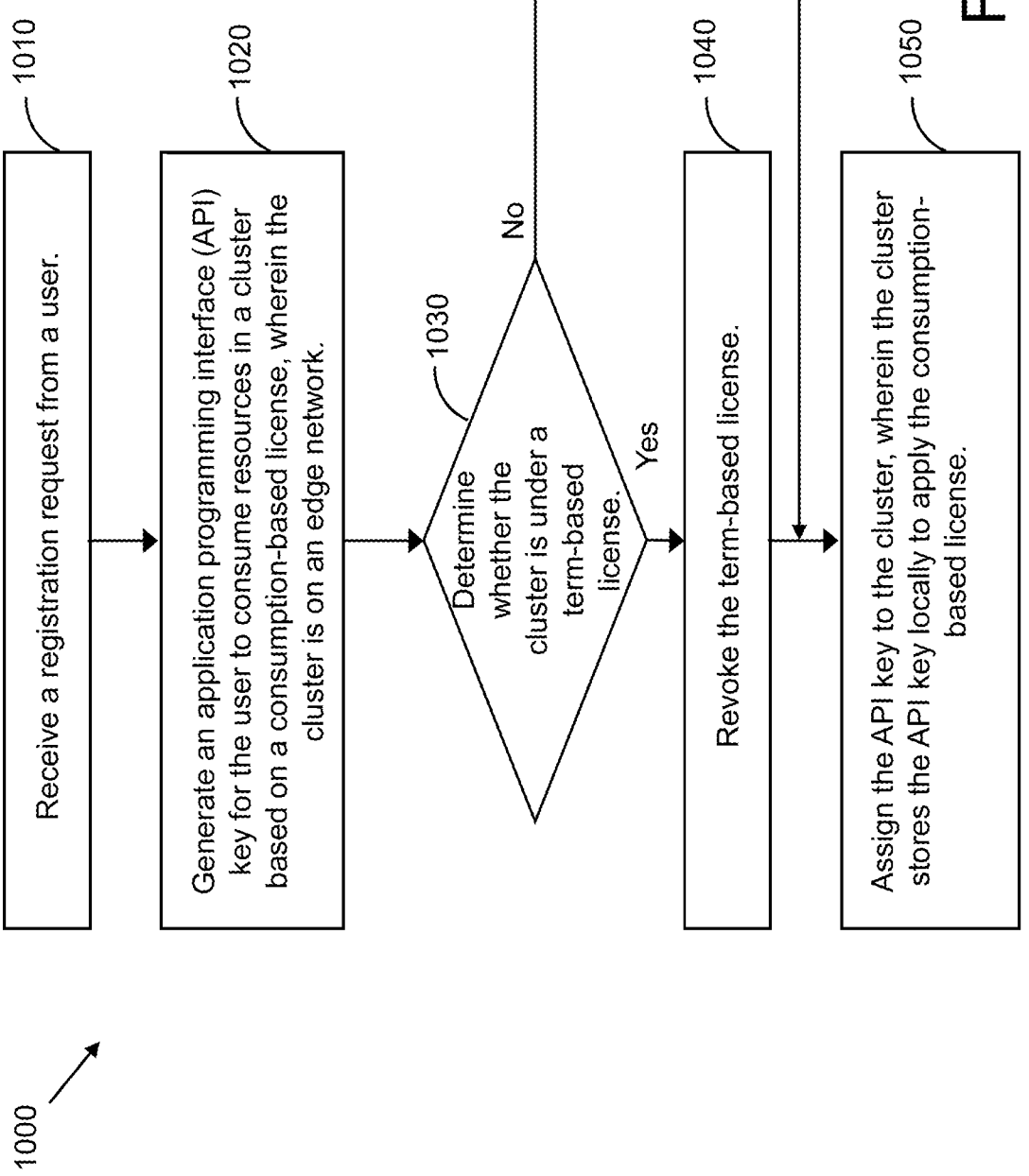

1000
1010
Receive a registration request from a user.
1020
Generate an application programming interface (API) key for the user to consume resources in a cluster based on a consumption-based license, wherein the cluster is on an edge network.
1030
Determine whether the cluster is under a term-based license.
Yes
No
1040
Revoke the term-based license.
1050
Assign the API key to the cluster, wherein the cluster stores the API key locally to apply the consumption-based license.

FIG. 10

# SYSTEM AND METHOD FOR RECONCILING CONSUMPTION DATA

## CROSS-REFERENCE TO RELATED APPLICATION

**[0001]** This application is related to and claims priority under 35 U.S. § 119(b) the Indian Patent Application No. 202141024135, filed May 31, 2021, titled "SYSTEM AND METHOD FOR SERVICE PROVIDER CONSUMPTION MODEL," the entire contents of which are incorporated herein by reference for all purposes.

## BACKGROUND

**[0002]** Virtual, containerized, and microservice oriented computing systems are widely used in a variety of applications. The computing systems include one or more host machines running one or more entities (e.g., workloads, virtual machines, containers, and other entities) concurrently. Modern computing systems allow several operating systems and several software applications to be safely run at the same time, thereby increasing resource utilization and performance efficiency. However, the present-day computing systems have limitations due to their configuration and the way they operate.

## SUMMARY

**[0003]** Various embodiments disclosed herein are related to a non-transitory computer readable storage medium. In some embodiments, the medium includes instructions stored thereon that, when executed by a processor, cause the processor to receive, at a server, from a first cluster of nodes on an edge network in communication with the server, first resource consumption data of a first service hosted on the edge network, calculate a first resource consumption quantity based on the first resource consumption data, receive, from a second cluster of nodes on the edge network, delayed resource consumption data of a second service hosted on the edge network, and calculate a second resource consumption quantity based on the delayed resource consumption data. In some embodiments, the first resource consumption data is collected at a first time and the delayed resource consumption data collected at the first time.

**[0004]** Various embodiments disclosed herein are related to an apparatus including a processor and a memory. In some embodiments, the memory includes programmed instructions that, when executed by the processor, cause the apparatus receive, at a server, from a first cluster of nodes on an edge network in communication with the server, first resource consumption data of a first service hosted on the edge network, calculate a first resource consumption quantity based on the first resource consumption data, receive, from a second cluster of nodes on the edge network, delayed resource consumption data of a second service hosted on the edge network, and calculate a second resource consumption quantity based on the delayed resource consumption data. In some embodiments, the first resource consumption data is collected at a first time and the delayed resource consumption data collected at the first time.

**[0005]** Various embodiments disclosed herein are related to a computer-implemented method. In some embodiments, the method includes receiving, at a server, from a first cluster of nodes on an edge network in communication with the server, first resource consumption data of a first service hosted on the edge network, calculating a first resource consumption quantity based on the first resource consumption data, receiving, from a second cluster of nodes on the edge network, delayed resource consumption data of a second service hosted on the edge network, and calculating a second resource consumption quantity based on the delayed resource consumption data. In some embodiments, the first resource consumption data is collected at a first time and the delayed resource consumption data collected at the first time.

**[0006]** The foregoing summary is illustrative only and is not intended to be in any way limiting. In addition to the illustrative aspects, embodiments, and features described above, further aspects, embodiments, and features will become apparent by reference to the following drawings and the detailed description.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0007]** FIG. **1** is an example block diagram of a computing environment for metering consumption, in accordance with some embodiments of the present disclosure.

**[0008]** FIG. **2**A is an example block diagram of the cluster of FIG. **1**, in accordance with some embodiments of the present disclosure.

**[0009]** FIG. **2**B is an example block diagram of the edge network of FIG. **1** that includes a super-cluster, in accordance with some embodiments of the present disclosure.

**[0010]** FIG. **3**A is an example block diagram of the metering service of FIG. **1**, in accordance with some embodiments of the present disclosure.

**[0011]** FIG. **3**B is an example block diagram of a computing environment including the metering service of FIG. **1**, in accordance with some embodiments of the present disclosure.

**[0012]** FIG. **4** is an example block diagram of a computing environment that includes a validation service, in accordance with some embodiments of the present disclosure.

**[0013]** FIG. **5** is an example flowchart of a method or metering resource consumption, in accordance with some embodiments of the present disclosure.

**[0014]** FIG. **6** is an example flowchart of a method for collecting resource consumption data, in accordance with some embodiments of the present disclosure.

**[0015]** FIG. **7** is an example flowchart of a method for updating resource consumption, in accordance with some embodiments of the present disclosure.

**[0016]** FIG. **8** is an example flowchart of a method for providing alerts, in accordance with some embodiments of the present disclosure.

**[0017]** FIG. **9** is an example flowchart of a method for validating a metering system, in accordance with some embodiments of the present disclosure.

**[0018]** FIG. **10** is an example flowchart of a method for registering a cluster under the consumption-based license model, in accordance with some embodiments of the present disclosure.

**[0019]** The foregoing and other features of the present disclosure will become apparent from the following description and appended claims, taken in conjunction with the accompanying drawings. Understanding that these drawings depict only several embodiments in accordance with the disclosure and are therefore, not to be considered limiting of

its scope, the disclosure will be described with additional specificity and detail through use of the accompanying drawings.

## DETAILED DESCRIPTION

**[0020]** In the following detailed description, reference is made to the accompanying drawings, which form a part hereof. In the drawings, similar symbols typically identify similar components, unless context dictates otherwise. The illustrative embodiments described in the detailed description, drawings, and claims are not meant to be limiting. Other embodiments may be utilized, and other changes may be made, without departing from the spirit or scope of the subject matter presented here. It will be readily understood that the aspects of the present disclosure, as generally described herein, and illustrated in the figures, can be arranged, substituted, combined, and designed in a wide variety of different configurations, all of which are explicitly contemplated and made part of this disclosure.

**[0021]** A consumption-based license model enables service providers to pay for consumption of infrastructure such as utilization or provisioning of resources for the service providers, which contrasts with a term-based license model in which the service providers pay a fixed amount for a time period regardless of how much resources they consume during that time period. In the consumption-based license model, the service providers can be metered for their consumption of resources/services in terms of networking, software as a service (Saas), platform as a service (PaaS), disaster recovery as a service (DraaS), infrastructure as a service (laaS), or many other services.

**[0022]** In some embodiments, other on-premises based solutions meter or charge the resource consumption locally for that particular deployment (mostly based on selling terms & conditions). In some embodiments, other solutions only offer public cloud consumption-based modeling because providers of such solutions have not overcome challenges of collecting consumption data at edge networks (e.g., private data centers or other public clouds separate from the public cloud metering the consumption) and aggregating different substrates into a single solution. What is needed is a unification capability that deploys the solution irrespective of customer site location.

**[0023]** Disclosed herein are some embodiments of a system, an apparatus, and a method for gathering the consumed resources from edge nodes (e.g., hyperconverged infrastructure, or HCl, nodes that provide computing, storage, and networking resources) which are running in the customer's on-premises, residing in a data center across the globe, or running on top of public cloud infrastructure, and sending the gathered data to centralized location where a metering service processes the data as per business requirements.

**[0024]** In some embodiments, the system, apparatus, and method guarantee that irrespective of the underlying substrate, the metering model is capable enough to calculate the resource consumption uniformly and single invoice generation becomes very easy. In some embodiments, the system, apparatus, and method ensure that gathered utilization data at source clusters (irrespective of its physical geographic location) is sent to the centralized location where the metering service has logic to identify and calculate each cluster's consumption data independently. In some embodiments, having such capability gives the system flexibility to apply different metering policies as per the cluster's substrate. For example, the system can charge the customer's resource consumption differently if it runs on a different substrate, as the operating cost varies per substrate, but keep the invoice generation common with a single centralized billing and policy managing solution.

**[0025]** In some embodiments, other solutions collect, filter, and process resource consumption data at a centralized server. Such solutions can use up a prohibited amount of network bandwidth to transmit the resource consumption data from the edge nodes where the consumption is happening to the server where the collecting is happening. Moreover, such solutions can overburden processors of the server because of the processing required to format the resource consumption data into a form that can be metered and billed.

**[0026]** Disclosed herein are some embodiments of a system, an apparatus, and a method for collecting at the edge nodes. In some embodiments, the collectors periodically gather the utilization data from the cluster and send a compact version of the utilization data to the centralized distributed system for analysis. In some embodiments, there are dedicated collectors for each supported service in the cluster. In some embodiments, collectors collect the resource utilization data at fine-level (e.g., minute level) granularity. Advantageously, this can allow customers to capture the resource consumption on a (substantially) real-time basis.

**[0027]** In some embodiments, data gathering happens at source clusters and data analysis happens at common centralized location. Beneficially, keeping the data gathering and data analysis apart can provide the flexibility of maintaining them separately without having any tight dependency on each other. Moreover, by gathering at the source, an amount of processing at the centralized location can be reduced.

**[0028]** The edge network can have multiple node clusters running in one or multiple substrates and each node of the cluster can capture the resource utilization in a distributed form. If one of the nodes of the cluster fails at the source, then one node's data can be retrieved from another node of the cluster. Advantageously, in some embodiments, the system prevents no data loss and metering of the cluster won't get affected in this scenario.

**[0029]** In some embodiments, if sending to the centralized system fails due to any reason, then this data is automatically stored in the clusters locally. For example, if the cluster comes up from a temporary failure or downtime, then the server checks what was the last successful data sent timestamp and from where to continue. In some embodiments, in the event of a network communication failure, the collectors persist the resource utilization data locally on the cluster and when the network communication is restored, they send all the accumulated data to centralized servers for data analysis.

**[0030]** In some embodiments, the system, apparatus, and method provide the flexibility of updating the collectors independent of underlying substrate and software. In some embodiments, the collectors are a microservice and utilization data gathering logic can be modified, maintained, and upgraded separately, remotely, and/or automatically, depending on the customer's requirements. In some embodiments, no major maintenance cycle is required to upgrade the collectors.

**[0031]** In some embodiments, some nodes or clusters are affected by a temporary network outage and the system incorrectly meters and charges for that cluster for that

network downtime. What is needed is a mechanism to correct such metering and billing errors.

**[0032]** Disclosed herein are some embodiments of a system, an apparatus, and a method for handling the under/over metering and charging use-cases effectively. To handle this case, in some embodiments, the metering service runs two tasks/check-pointers (e.g., a regular task and a fixer task). The metering service, in executing a regular task, can calculate the last one hour of consumption data received from source clusters. The fixer task can be executed later than the regular task (e.g., a day later, can be configured). In some embodiments, the metering service, in executing the fixer task, again calculates the consumption for the same time period for which the regular task was executed. In some embodiments, if the system finds a difference in the calculation, the system updates the previous calculation. Advantageously, the fixer task can reconcile utilization data for inadvertent network failure and for dark-sites, in which network availability is limited by design. Moreover, recalculating resource consumption can be used for consumption data auditing.

**[0033]** The metering service may face missing resource consumption data, delayed resource consumption data, and/or application programming interface (API) connectivity issues for registration or billing. What is needed is a mechanism for alerting a user or a site reliability engineer (SRE) of these issues so they can be addressed.

**[0034]** Disclosed herein are some embodiments of a system, an apparatus, and a method for alerting customers or SREs of data or API issues that affect the correctness of the calculated resource consumption and/or billable amount. In some embodiments, the system receives an indication that data is missing or delayed. In some embodiments, the system receives an indication that an API is not reachable. In some embodiments, the system alerts the customers or SREs based on one of the indications. Advantageously, based on the alert, the user or SRE can manually intervene, e.g., by configuring another fixer task so that the metering service can meter the delayed data.

**[0035]** In some embodiments, the consumption metering may encounter errors for certain use-cases. What is needed is a tool that can serve as a commercial-product for service providers to validate metering.

**[0036]** Disclosed herein are some embodiments of a system, an apparatus, and a method for validating metering for consumption-based offering for service providers. In some embodiments, the system validates full depth & breadth of the consumption-based licensing for service providers. In some embodiments, the system covers all the basic use-cases from an end-to-end perspective that any service provider would like to validate upon debugging product failure. Advantageously, validating metering can improve robustness of the system for various use-cases.

**[0037]** Other solutions require manually copying a cluster configuration from the cluster and manually uploading the cluster configuration onto a registration service/portal before acquiring a license key from the portal which is uploaded to the cluster. What is needed is a more automated approach for registering a cluster for a consumption-based license.

**[0038]** Disclosed herein are some embodiments of a system, an apparatus, and a method for registering in a more automated way. In some embodiments, the system generates an API key and assigns the API key to a cluster registered to the user. The license is applied when the API key is stored in the cluster. Advantageously, some embodiments skip the step of having to download the cluster configuration from the cluster and upload it to the registration service, resulting in a better user experience.

**[0039]** FIG. **1** illustrates an example block diagram of a computing environment **100** for metering consumption, in accordance with some embodiments of the present disclosure. The computing environment **100** includes a server **105**. In some embodiments, the server **105** is a centralized server or a distributed server. The server **105** is coupled to (e.g., in communication with) an edge network **110**. The server **105** processes information received from the edge network **110**.

**[0040]** The computing environment **100** includes a registration service (e.g., a customer portal) **115**. The registration service **115** can be hosted by the server **105** or a server/node/VM/container separate from the server **105**. The registration service **115** registers a user (e.g., a tenant, customer, service provider, service provider's end user, etc.) or a device associated with the user for consuming cluster resources on a consumption-based license model. The user can request consumption-based registration/licensing of new clusters or existing clusters (e.g., transitioning from another license model such as a term-based license model).

**[0041]** In some embodiments wherein new clusters are being requested, the registration request includes one or more of a number of clusters, a number of nodes on each cluster, types of services to be registered (e.g., in each of the clusters), a number of super-clusters (e.g., multi-cluster management services), etc., from the user. In some embodiments, the registration request includes types of services to be registered and automatically determines a number of clusters and a number of nodes based on the service requirement. In some embodiments, the registration service **115** registers the services in the respective nodes and clusters in accordance with the request. In some embodiments, the registration service **115** assigns a user ID (e.g., tenant ID, user account, tenant account) for the user associated with the cluster and/or a cluster ID for each of the clusters to be registered. In some embodiments, the clusters to be registered are dedicated to that user (e.g., cluster per tenant) whereas in other embodiments, the clusters to be deployed are shared with other users (e.g., multi-tenant clusters).

**[0042]** In some embodiments wherein a user is transitioning to the consumption-based license, the registration service **115** receives, from the user, the user ID corresponding to the user and/or each of the cluster IDs associated with the (respective) clusters corresponding to the user under another license model. In other embodiments of a transitioning user, the registration service **115** receives the cluster information (e.g., number of clusters, number of nodes, types of services, etc.) and assigns the user ID and/or the cluster IDs. In some embodiments, the user has pending/not-yet-used credit with the term-based license that is transferred to the consumption-based license. The credit may be used to pay for resource consumption equal to a value of the credit.

**[0043]** In some embodiments, in response to receiving the registration request, the registration service **115** generates a token (e.g., an application programming interface (API) key, a file) e.g., for the user to consume resources based on the consumption-based license (model). In some embodiments, the token is per-user or per-cluster. In some embodiments, the registration service **115** assigns the token to the registered cluster (e.g., the cluster **120**A on the edge network **110**) associated with the user. In some embodiments, the user

copies the API key from the registration service and stores the API key in the registered.

**[0044]** In some embodiments, the token includes one or more of a user ID, a cluster ID, or a policy ID. In some embodiments, the registration service **115** assigns a token for each cluster and each super-cluster. The token may be stored in memory or storage in the server **105** or the network. In some embodiments, by storing the token, the license/token is applied to the cluster where the token is stored. In some embodiments, once the token is applied, the server **105** can start receiving collected consumption data from each cluster and metering consumption of services on each cluster and the registration service **115** can pull information from the registered cluster. After the license is applied, the resource consumption data (e.g., input to metering) and/or the metering data (e.g., output from metering) includes the user ID that allows matching resource consumption/metering data with a correct user. In some embodiments, matching the data to a user eliminates or reduces a potential of mismatch of data and a user.

**[0045]** In some embodiments, after the license is applied, the user (e.g., via the registration service **115** or a UI of the cluster) can scale up or scale down the cluster configuration without having to change or move the token or increase/decrease the number of tokens. For example, the user adds nodes or removes nodes from the cluster; the user changes the operating system, or aspects thereof, (e.g., usage tier) from a first type (e.g., without additional features) to a second type (e.g., with additional features); the user increases or decreases an amount of memory/storage (e.g., non-volatile memory such as flash which can include solid-state drive (SSD) or NVM express (NVMe)), a number of file servers or stored data, a number of object stores, a number of nodes protected by a security security, or a number of VMs to be used by a service.

**[0046]** In some embodiments, the registration service **115** de-registers a user (e.g., upon request of the user). In some embodiments, de-registering a user includes stopping metering of services that on the cluster, stopping sending of metered data, removing the token from the cluster (e.g., user can transition to a term-based license), and marking the cluster as inactive.

**[0047]** The edge network **110** is or includes one or more of an on-premises data center, a distributed data center (e.g., a third-party data center, a data center that serves an enterprise), a private cloud, or a public cloud (e.g., different from a public cloud that hosts the server **105**). The edge network **110** includes a number of clusters **120**A, **120**B, . . . , **120**M. The cluster **120**A includes a number of services **125**A, **125**B, . . . , **125**N. Each of the number of services **125**A, **125**B, . . . , **125**N can be a different service. For example, the service **125**A may include one or more of an operating system/kernel/core service, a user interface, database provisioning, lifecycle management, orchestration/automation, networking security (e.g., micro-segmentation of the network), a (e.g., software-defined) file server, etc. In some embodiments, each of the services **125**A-**125**N include, correspond to, or are coupled to a respective collector **130** that collects data/metadata such as resource utilization/consumption from each of the services **125**A-**125**N. In other embodiments, the services **125**A-**125**N are coupled to a single collector.

**[0048]** Each of the services **125**A-**125**N may be running/executed on a virtual machine (VM) or container. Although the disclosure focuses on the cluster **120**A and the service **125**A, any of the clusters **120**B-**120**M and any of the services **125**B-**125**N are within the scope of the disclosure. Although FIG. **1** shows three clusters **120**A-**120**N and three services **125**A-**125**N, any number of clusters and services are within the scope of the disclosure.

**[0049]** FIG. **2**A is a more detailed, example block diagram of the cluster **120**A of FIG. **1**, in accordance with some embodiments of the present disclosure. The cluster **120**A includes the number of services **125**A-**125**N. Each service consumes resources from nodes. As an example, the service **125**A consumes resources from the nodes **206**A, **206**B, . . . , **206**K. Each node includes resources. As an example, the node **206**A includes resources **208** which include CPU (cores) **210**, memory **212**, NICs (and other networking resources) **214**, and storage **216**. The resources **208** are provided to the service **125**A via the virtualization (layer, e.g., hypervisor or container runtime) **218**. In some embodiments, the node **206**A is referred to as a hyperconverged infrastructure (HCl) node because the node **206**A provides the CPU cores **210**, the memory **212**, the NICs **214**, and the storage **216** resources, as opposed to a three-tier architecture which segregates different types of resources into different nodes/servers/etc. In some embodiments, the cluster **120**A is referred to as an HCl cluster.

**[0050]** Each of the services **125**A-**125**N includes a consumption collector **220**. The consumption collector **220** collects service resource consumption data **222** (e.g., information, files, statistics, metadata, etc.). In some embodiments, the service resource consumption data **222** indicates resource consumption of the respective service (e.g., that the consumption collector **220** is running on or corresponds to). In some embodiments, the service resource consumption data **222** includes an identifier of the resource, a time stamp (indicating a time), and a consumption amount corresponding to the resource. For example, the consumption data can include "VM**1** **10**:**30**AM **4**GB." The time, the amount, and the identifier may be referred to as a consumption data point. In some embodiments, the service resource consumption data **222** includes a plurality of consumption data points. In some embodiments, the service resource consumption data **222** includes a user ID of the user consuming the resources. In some embodiments, the service resource consumption data **222** includes a state of the respective service (e.g., powered on or off). In some embodiments, the consumption collector **220** is similar to the collector **130** of FIG. **1**. Each of the services **125**A-**125**N may include other collectors such as log collectors, configuration collectors, health collectors, etc.

**[0051]** In some embodiments, the cluster **120**A includes an aggregate collector **224** that is in communication with each consumption collector **220**. In some embodiments, the aggregate collector **224** aggregates the service resource consumption data **222** of all of the consumption collectors to provide a cluster resource consumption data **226** which indicates resource consumption at the cluster level. In some embodiments, the aggregate collector **224** specifies/defines a frequency of collection and an amount/limit of data to aggregate into one collection/set of data. In some embodiments, the aggregate collector **224** retains service resource consumption data **222** and filters out some or all other types of data (e.g., cluster/service health data).

**[0052]** In some embodiments, the cluster **120**A includes a cluster repository **228**. The aggregate collector **224** stores

the cluster resource consumption data **226** in the cluster repository **228**. In some embodiments, the cluster repository **228** is in-memory. In some embodiments, the cluster repository **228** is, or includes, one or more of log-based storage or a relational database.

**[0053]** In some embodiments, the cluster **120**A includes a collector frame service (CFS) **236**. The CFS **236** may receive the cluster resource consumption data **226** from the cluster repository **228** and provides a second (e.g., buffered) cluster resource consumption data **238** to the server **105**. In some embodiments, the buffered cluster resource consumption data **238** is similar to the cluster resource consumption data **226**. In some embodiments, the buffered cluster resource consumption data **238** is formatted in a way that can be interpreted by the server **105**. In some embodiments, the buffered cluster resource consumption data **238** includes additional consumption data such as consumption data of services external (e.g., running on top of) the cluster **120**A. The CFS **236** may perform various other functions such as instructing one or more of the collectors **220** or **224** to change a configuration, identifying false positives, add one or modify rules to correct for errors and false positives, provide or resolve conflicts of override configuration rules, etc. In some embodiments, the collector configuration includes one or more of what information to collect, where to collect the information from, how to collect the information, how granular to collect this information, when to collect this information, how often to collect, and when and where to push the information.

**[0054]** In some embodiments, the server **105** determines that the cluster **120**A (e.g., ora service, e.g., the service **125**A, or a node hosting the service, e.g., the node **206**A hosting the service **125**A) is powered off if the cluster **120**A is temporary or permanently failing/down, which can be referred to as a source failure, or the user has configured the cluster **120** to be powered down. In some embodiments, the server **105** determines that the cluster **120**A is powered on if communication (e.g., a network, a link, etc.) between the cluster **120**A and the server **105** is down/terminates/interrupts or if the cluster **120**A is in a dark-site state (e.g., intentionally not communicating with the server **105** for privacy purposes, etc.). In some embodiments, during the outage, each consumption collector **220** persists/stores the service resource consumption data **222** of the respective service locally in the cluster repository **228** and the CFS **236** sends the buffered cluster resource consumption data **238** (e.g., the service resource consumption data **222** for the current time period and for the time period in which there was in an outage) after communication with the server **105** reestablishes/resumes/is restored.

**[0055]** In some embodiments, the server **105** determines that the failure is a source failure by (a) not receiving the buffered cluster resource consumption data **238** (e.g., within/for a certain time period), but (b) receiving indication that communication with the edge network **110** is active/uninterrupted (e.g., receiving a success code/response/acknowledgment in response to a health/polling/status query/request). In some embodiments, the server **105** determines that the failure is a network failure (e.g., a failure of a communication network in between the edge network **110** and the server **105**) by (a) not receiving the buffered cluster resource consumption data **238** (e.g., within/for a certain time period), and (b) receiving indication that communication with the edge network **110** is inactive/interrupted (e.g., receiving a failure code/response/non-acknowledgement in response to the health query). In some embodiments, the server **105** determines a duration of no data being (successfully) sent (e.g., based on timestamps of data successfully being sent).

**[0056]** FIG. **2**B is a more detailed example block diagram of the edge network **110** of FIG. **1** that includes a super-cluster **240**, in accordance with some embodiments of the present disclosure. Although only one cluster is shown, the edge network **110** can include two or more clusters that are coupled to the super-cluster **240**. The super-cluster **240** aggregates data from one or more clusters such as the cluster **120**A and one or more external services **230**A, **230**B, . . . **230**J. In some embodiments, the external services **230**A-**230**J are services that are associated with the user and the consumption license but that are not included in any of the clusters communicating with the super-cluster **240**. In some embodiments, the external services **230**A-**230**J are running on third-party infrastructure. In some embodiments, each of the external services **230**A-**230**J includes one or more collectors such as the consumption collector **220**. In some embodiments, each of the external services **230**A-**230**J are similar to a respective one of the services **125**A-**125**N of FIG. **1**.

**[0057]** The super-cluster **240** includes a super-cluster repository **232**. The super-cluster repository **232** receives the cluster resource consumption data **226** from each data repository such as the cluster repository **228** and from the external services **230**A-**230**L. In some embodiments, the cluster resource consumption data **226** is received at a predetermined interval.

**[0058]** In embodiments corresponding to FIG. **2**B, the super-cluster **240** includes a super-cluster collector **234** and the CFS **236** (and the CFS **236** is omitted from the cluster **120**A). The super-cluster collector **234** fetches the aggregated data from the super-cluster repository **232**. The super-cluster collector **234** may perform similar functions as the aggregate collector **224**. In some embodiments, the super-cluster collector **234** provides the collected data to the CFS **236**. The CFS **236** may generate data similar to the buffered cluster resource consumption data **238** based on the aggregate data received from the super-cluster collector **234**.

**[0059]** Returning to FIG. **1**, the server **105** includes a data processing pipeline **135** that receives the data collected by each collector such as the collector **130**. In some embodiments, the data processing pipeline **135** performs schema validation, converts (e.g., aggregates, formats) the buffered cluster resource consumption data **238** received from different devices and services into a detailed metering item **142**. In some embodiments, the detailed metering item **142** includes one or more of a user ID, a resource/entity ID, a resource consumption amount/quantity (e.g., at a cluster level), a region, a policy ID, a duration, supported attributes of the cluster or service therein, a service that consumed the resource, or a (power) state of the service. In some embodiments, the detailed metering item **142** is a Javascript object notation (JSON) stream. In some embodiments, the data processing pipeline **135** persists/stores the detailed metering item **142** in a data repository (e.g., data lake, database, etc.) **140**. In some embodiments, the server **105** includes the data repository **140**.

**[0060]** The server **105** includes a metering service **145** in communication with the data repository **140**. In some embodiments, the metering service **145** receives the detailed

metering item **142** from the data repository **140**. In some embodiments, the metering service **145** converts/transforms/formats the detailed metering item **142** into a charge item **148**. In some embodiments, the charge item **148** is at a user level. The metering service **145** may aggregate consumption of different services **125**A-**125**N or different clusters **120**A-**120**M to a user level of consumption. In some embodiments, the charge item **148** includes one or more of the user ID, a duration (e.g., a start time and a stop time), a unit of measurement (UoM), a quantity (e.g., in terms of the UoM), or a region. The UoM may include one or more of a resource type (e.g., one or more resources such as central processing unit (CPU) cores (e.g., VMs, containers), storage (e.g., disks), or memory) or a time granularity/unit/interval for quantifying resource consumption (e.g., minute, hour, day). In some embodiments, the charge item **148** is calculated or formatted according to one or more metering policies, which is discussed below in more detail.

**[0061]** In some embodiments, the server **105** includes a metering storage (e.g., database) **150** in communication with the metering service **145**. In some embodiments, the metering service **145** stores the output state (e.g., the charge item **148**) and a detailed split up of usage (e.g., the detailed metering item **142**) in the metering storage **150**. In some embodiments, the metering service **145** pulls user license information (e.g., a list of the clusters that the user registered, metering policies, etc.) from the registration service **115** periodically and persists the user license information into the metering storage **150**. In some embodiments, the metering service **145** persists metering policies in the metering storage **150**. In some embodiments, the metering service **145** persists a metadata state into the metering storage **150** (e.g., for bootstrapping after restarts and for debuggability, etc.). In some embodiments, the metadata state which is captured includes a task/user execution state along with relevant checkpoints with respect to task execution, each task's state (success/failure, execution latency etc.).

**[0062]** FIG. **3**A is a more detailed example block diagram of the metering service **145**, in accordance with some embodiments of the present disclosure. In some embodiments, the metering service **145** is a (e.g., containerized) microservice. The metering service **145** includes a metering master (e.g., master) **305** and a number of metering workers (e.g., workers) **310**A, **310**B, . . . , **310**L. In some embodiments, the metering master **305** and the metering workers **310**A-**310**L are microservices or threads of a single microservice. In some embodiments, instances of the metering master **305** and the metering workers **310**A-**310**L can be deployed in individual groups/pods including shared storage, networking, and instructions for how to run the metering master **305** and the metering workers **310**A-**310**L such as an image of each of the metering master **305** and the metering workers **310**A-**310**L and ports to use. In some embodiments, the metering master **305** and the metering workers **310**A-**310**L are deployed as VMs or containers using a VM deployment platform or container deployment platform, respectively. Each service can scale up and down according to a workload and achieve a high-level of reliability.

**[0063]** In some embodiments, the metering master **305** schedules tasks for the workers **310**A-**310**L. The metering master **305** can be responsible for bootstrapping the metering state (e.g., a list of users, checkpoints, policies) from a persistent store upon start. In some embodiments, the metering master **305** provides/fronts public-facing metering APIs for retrieving the metering output state (e.g., the charge item **148**, a user/task metadata state, detailed records/charge items for a user, detailed metering item **142**).

**[0064]** In some embodiments, the metering master **305** pulls (e.g., retrieves, fetches) user license/registration information from the registration service **115** (e.g., which users are registered under the consumption-based license models) periodically (e.g., by polling the registration service **115**) and persists the user license information into a metering database. The registration service **115** exposes an API to query a current list of registered users. The metering master **305** can use a hypertext transfer protocol (http) request with a proper user/bearer token to communicate with the registration service **115** API.

**[0065]** Each of the metering workers **310**A-**310**L is responsible for executing one or more tasks. As an example, the metering worker **310**A executes one or more metering tasks. In some embodiments, the metering worker **310**A pulls one or more metering tasks from the metering master **305** and calculates the resource consumption for the given unit of measure (UoM), the user, and the duration/number, based on a selected metering policy of one or more metering policies **315**A, **315**B, . . . , **314**P. In some embodiments, the metering worker **310**A uses a policy ID provided in the metering task to retrieve a metering policy from the metering storage **150** (e.g., by finding the metering policy or an address thereof at an index equal to the policy ID or hash of the policy ID). In some embodiments, the metering worker **310**A determines the UoM from contents of the retrieved metering policy. The metering worker **310**A can process tasks in a number of concurrent threads for execution (e.g., as a command-line flag). Each of the metering workers **310**A-**310**L can scale independently by having multiple processes. Although the disclosure focuses on the metering worker **310**A, any of the metering workers **310**B-**310**L are within the scope of the disclosure. Although FIG. **3**A shows three metering workers **310**A-**310**L, any number of metering workers are within the scope of the disclosure.

**[0066]** In some embodiments, a metering task includes/encapsulates one or more of user information (e.g., user ID), a policy ID, Start/End time, a type of task, a created timestamp, and once executed, also holds info for the status and task execution time. In some embodiments, the metering task can include information from the detailed metering item **142**. In some embodiments, the metering worker **310**A executes/runs a regular task (e.g., a pointer), which includes computing metering for the given user and duration to provide the charge item **148**. The regular task may be near the time (e.g., within one hour) the resource consumption data is used. The metering worker **310**A (or the metering master **305**) may save/buffer resource consumption data for a certain amount of time (e.g., one hour). In some embodiments, the regular task includes a time stamp that indicates up to what time metering has been performed on the resource consumption data.

**[0067]** In some embodiments, the metering worker **310**A executes a fixer task, which runs (e.g., based on a command-line flag such as a glfag) a certain time (e.g., hours, days) after a respective regular task and computes the metering again. The fixer tasks can serve as a safeguarding mechanism by accounting for late arrival of input data (e.g., input data that was collected before a corresponding regular task but not sent to the server **105** until after the corresponding

regular task) and outage of the one of the components of the edge network **110**, the server **105**, or a network coupling the edge network **110** and the server **105**.

**[0068]** In some embodiments, a time delta/delay between executing the regular task and the fixer task is preconfigured (e.g., by the server **105** or the user). In some embodiments, the time delta between the regular tasks and the fixer task is set/adjusted/modified (manually) by the user. In some embodiments, the fixer task can be executed more than once for a given user and duration (e.g., based on an alert, which is discussed in further detail below).

**[0069]** Since there are multiple users, multiple time slices (monthly, daily, hourly) and different kinds of tasks (e.g., regular and fixer), the metering master **305** can prioritize tasks. In some embodiments, the scheduler first schedules the regular tasks (e.g., in an order of highest granularity to lowest granularity, such as monthly, daily, hourly) before scheduling to fixer tasks. In some embodiments, the task execution is idempotent (e.g., any task from any time can be executed again without corrupting any of the internal metadata state or output, which are both persisted in a metering database, or a packet sent to a billing service).

**[0070]** In some embodiments, the metering policy **315**A includes user-defined rules that specify how to meter a given resource/entity for a given one or more users. In some embodiments, upon defining/receiving a policy, the computing environment **100** (e.g., the registration service **115**, the metering service **145**) applies the metering policy **315**A to the applicable users. In some embodiments, the metering policy **315**A includes the UoM (e.g., a resource to be metered, time ranges for the computation), attribute names and properties (e.g., which attributes to be considered for that type of resource and other specific properties on how to use that attribute), specific calculation methods to be applied, time ranges for reporting, complementary and discount services, and other miscellaneous support attributes. In some embodiments, the metering worker **310**A receives the metering policy **315**A as part of the task or receives it separately from the metering master **305** or a database. Although the disclosure focuses on the metering policy **315**A, any of the metering policies **315**B-**315**P are within the scope of the disclosure. Although FIG. **3**A shows three metering policies **315**A-**315**P, any number of metering policies are within the scope of the disclosure.

**[0071]** In some embodiments, the UoM (e.g., a charge item, a granularity, a time granularity, a combination of a granularity and a resource type, a number of resources, etc.) varies based on a service used. For example, a first UoM and a second UoM for an operating system service are number of CPU core hours and number of (flash) memory hours, a third UoM for a user interface (UI) service is a number of nodes, a fourth UoM for an orchestration/automation service is a number of VMs, and a fifth UoM for a file server and for an object store is an amount of stored Tebibytes (TiB). In some embodiments, if the resource/UoM is or corresponds to a compute resource, the metering worker **310**A computes the resource consumption only for when the service using the resource is powered on, whereas if the resource/UoM is or corresponds to a storage resource, the metering worker **310**A computes the resource consumption regardless of whether the service using the resource is powered on or powered off.

**[0072]** FIG. **3**B is an example block diagram of a computing environment **300**, in accordance with some embodiments of the present disclosure. In some embodiments, the computing environment **300** is similar to the computing environment **100** of FIG. **1**. However, for the purpose of showing how the metering service **145** interacts with other components, the computing environment **300** illustrates more details in some aspects and omits details in other aspects with respect to the computing environment **100** of FIG. **1**.

**[0073]** In some embodiments, the metering master **305** receives license information **320** from the registration service (e.g., clusters and/or services registered, metering policies). The license information **320** may be sent in snapshots. The metering master **305** may poll the registration service **115** at a certain interval (e.g., **5** minutes) to receive the license information **320**. In some embodiments, the cluster **120**A provides the buffered cluster resource consumption data **238**, including the resource consumption of services at a cluster level and the policy ID, to the server **105**. In some embodiments, the data processing pipeline **135** converts the buffered cluster resource consumption data **238** into the detailed metering item **142** and provides the detailed metering item **142**, including the resource consumption of services at a cluster level and the policy ID, to the metering master **305**. In some embodiments, the metering master **305** polls the cluster **120**A at a certain interval, while in some other embodiments, the cluster **120**A provides the detailed metering item **142** at an certain interval or in response to a change in resource consumption without being polled. In some embodiments, the metering master **305** stores the license information **320** and the detailed metering item **142** in the metering storage **150**.

**[0074]** In some embodiments, the metering master **305** sends a task **325**, including instructions for executing the task **325**, to a metering worker **310**A. In some embodiments, the metering worker **310**A uses the policy ID to retrieve the metering policy **315**A from the metering storage **150**. In some embodiments, the metering worker **310**A executes the task **325** according to the instructions in the task **325**. In some embodiments, the metering worker **310**A computes or generates the charge item **148** based on the task **325** and the metering policy **315**A. For example, the metering policy **315**A specifies to compute a number of VM-hours and the task **325** specifies that cluster **120**A consumed 2 VMs for 30 minutes, 4 VMs for 30 minutes and 5 VMs for 1 hour. In the example, the metering worker **310**A computes the VM-hours, e.g., by normalizing each VM-hour data point to one hour and including the weight, multiplying the weight by the VM-hour data point, and adding the product together. In some embodiments, the metering worker **310**A provides the charge item **148** to the billing service **160**.

**[0075]** Returning to FIG. **1**, the server **105** includes an alerts service **155**. In some embodiments, the alerts service **155** determines or receives indication (e.g., from the metering service **145**) of one or more issues. For example, the metering service **145** persists a metric that indicates whether there is an issue (e.g., dataMissing=True/False) to the metering storage **150** and, if dataMissing=True, the metering service **145** provides an event to the alerts service **155**. In some embodiments, an issue is detected with respect to the entire cluster (e.g., if at least one service sends data, no issue is detected). In other embodiments, the issue can be detected with respect to resources, services, or policies.

**[0076]** In some embodiments, the issue includes one of data delay, data missing, or API connectivity issues. Data

delay can be when the cluster **120**A sends buffered cluster resource consumption data **238** after a regular task but within a predetermined delay threshold (e.g., 12 hours after a task). Data missing can be when the cluster **120**A does not send buffered cluster resource consumption data **238** within the predetermined delay threshold. In some embodiments, such as if the data delay or data missing is with respect to a regular task, the user can adjust the time that a fixer task is to run. In some embodiments, the user schedules another fixer task. The fixer task or the other fixer task can calculate and send an updated charge item such as the charge item **148**. In some embodiments, a site reliability engineer (SRE) manually calculates the updated charge item and posts it in the billing service **160**.

**[0077]** API connectivity issues can be when the metering service **145** cannot connect to the registration service **115** API to receive (e.g., a latest snapshot of) the license information **320** from the registration service **115**. In some embodiments, the metering service **145** polls the registration service **115** once per a certain interval. In some embodiments, if the metering service **145** does not receive the license information **320** from the registration service **115** after a predetermined number of intervals, the alerts service **155** receives an indication of an API connectivity issue.

**[0078]** API connectivity issues can be when the metering service **145** cannot connect to the billing service **160** API to provide the charge item **148** to the billing service **160**. In some embodiments, if posting the charge item **148** to the billing service **160** fails and/or a metering checkpoint fails, the alerts service **155** receives an indication of an API connectivity issue. In some embodiments, if the billing service **160** does not receive a request to post a charge item **148** for greater than a predetermined threshold for posting billing, the alerts service **155** receives an indication of an API connectivity issue. In some embodiments, a metering SRE or developer fixes/unblocks the connection that is causing the API connectivity issue.

**[0079]** In some embodiments, the alerts service **155** alerts/notifies a user or a site reliability engineer (SRE) of the issue. In some embodiments, the alerts service **155** generates or provides a corrective action. In some embodiments, the corrective action includes that the SRE manually fixes the issue, recalculates the charge item **148**, or tells the user what is wrong with the cluster. In some embodiments, the corrective action includes that the metering service **145** double-check a charge item **148** associated with the issue. If the issue is resolved within a predetermined resolution time, the metering service **145** can automatically updates the charge item **148**. If the issue is resolved after the predetermined resolution time, the SRE can manually recalculate and update the charge item **148**.

**[0080]** The computing environment includes a billing service **160** in communication with the metering service **145**. In some embodiments, once a task execution has been successfully completed, the metering service **145** provides/posts the charge item **148** (e.g., a packet, an output packet) to the billing service **160**. In some embodiments, the charge item **148** includes one or more of a user ID, a resource consumption quantity/value, a UoM, and a start and end date. In some embodiments, the charge item **148** is provided by or corresponding to execution of the respective task. In some embodiments, the billing service **160** multiplies the resource consumption quantity by a rate to determine a billable amount. In some embodiments, the rate is based on the metering policy **315**A. In some embodiments, the billing service **160** consolidates the formatted consumption data received from the metering service **145** into one data structure (e.g., spreadsheet, invoice, bill). In some embodiments, the billing service **160** sends, displays, or otherwise makes available the charge item **148** and the billable amount to the use (e.g., once per a certain interval). In some embodiments, the charge item **148** and the billable amount are displayed or otherwise represented versus time (e.g., time segments, time intervals).

**[0081]** FIG. **4** is an example block diagram of a computing environment **400** that includes a validation service **405**, in accordance with some embodiments of the present disclosure. In some embodiments, the computing environment **400** is similar to the computing environment **100** of FIG. **1**. However, for the purpose of showing how the validation service **405** interacts with other components, the computing environment **300** illustrates more details in some aspects and omits details in other aspects with respect to the computing environment **100** of FIG. **1**.

**[0082]** In some embodiments, the validation service **405** validates operations of one or more services related to metering resource consumption in a consumption-based license model (e.g., the registration service **115**, the data processing pipeline **135**, the metering service **145**, or the billing service **160**). Generally, the validation service **405** provides input data to one of the services, queries the one of the services, receives an actual response from the service based on the query, compares an actual response of the one of the services to an expected response (based on the input data), and validates the one of the services if the actual response matches the expected response. For example, the validation service **405** configures a cluster in the registration service **115** and queries the registration service **115** to determine if the configured cluster is registered.

**[0083]** In another example, the validation service **405** assigns a workload to a registered cluster (e.g., the cluster **120**A), wherein the validation service **405** knows a priori an amount of resources consumed, e.g., an amount of storage the workload is to consume (based on a size of the workload/file) or an amount of time that the CPU and/or memory the workload is to consume (based on a capacity of the CPU and/or memory and an amount of CPU and/or memory that is to complete the workload). In some embodiments, the validation service **405** queries the data processing pipeline **135**, the metering service **145**, or the billing service **160** to retrieve the amount of resources consumed. For example, the validation service **405** queries one or more of the data processing pipeline **135** to retrieve the buffered cluster resource consumption data **238** or the detailed metering item **142**, the metering service **145** to retrieve the detailed metering item **142** or the charge item **148**, or the billing service **160** to retrieve the charge item **148** or the billable amount.

**[0084]** Referring now to FIG. **5**, a flowchart of an example method **500** for metering resource consumption is illustrated, in accordance with some embodiments of the present disclosure. The method **500** can be performed by one or more systems, components, or modules depicted in FIGS. **1-4**, including, for example, the server **105**, the metering service **145**, etc. In some embodiments, instructions for performing the method **500** are executed by a processor included in, or associated with, the one or more systems, components, or modules and stored in a non-transitory computer readable storage medium included in, or associ-

ated with, the one or more systems, components, or modules. Additional, fewer, or different operations may be performed in the method **500** depending on the embodiment.

**[0085]** According to the method **500**, a processor (e.g., the metering service **145** or a processor therein) receives, at a server (e.g., the server **105**), from a cluster (e.g., the cluster **120**A) on an edge network (e.g., the edge network **110**) in communication with the server, resource consumption data (e.g., the buffered cluster resource consumption data **238**, the detailed metering item **142**, etc.) of a service (e.g., the service **125**A) hosted on the edge network (at operation **510**). In some embodiments, the resource consumption data includes one or more data points, and each data point includes a resource identifier, a time stamp, and a resource quantity. In some embodiments, the server is a first public cloud and the edge network or the cluster of nodes is, or is a portion of, one or more of an on-premises data center, a distributed (e.g., third-party) data center, a private cloud, or a second public cloud different from the first public cloud, or a combination thereof. In some embodiments, the resource consumption data is at a cluster level (e.g., takes into account resources consumed for the entire cluster).

**[0086]** In some embodiments, the processor receives, from a second cluster on the edge network, second resource consumption data of a service hosted on the edge network. In some embodiments, the cluster of nodes is on one type of platform and the second cluster of nodes is on another type of platform. For example, the cluster of nodes is on an on-premises data center and the second cluster of nodes is on a private cloud. Other examples of combinations of platforms are within the scope of the disclosure. In some embodiments, a user is registered with both of the cluster of nodes and the second cluster of nodes. In some embodiments, the processor generates the resource consumption quantity at least based on both of the resource consumption data and the second resource consumption data.

**[0087]** In some embodiments, the processor determines, based on one or more of a metering policy (e.g., the metering policy **315**A) or the resource consumption data, a unit of measurement (at operation **520**). In some embodiments, the unit of measurement includes a time granularity or a type of resource. In some embodiments, the processor calculates a resource consumption quantity (e.g., a charge item **148**) according to the unit of measurement (at operation **530**). In some embodiments, the resource consumption quantity is used to determine an amount (in dollars) to charge a user that is registered, or otherwise associated, with the cluster and any other clusters. In some embodiments, the resource consumption quantity is at a user level (e.g., takes into account resources consumed by the user regardless of the cluster).

**[0088]** In some embodiments, the processor determines cluster of nodes by retrieving license information associated with a user registered with the cluster of nodes from a registration service via a hypertext transfer protocol (HTTP) application programing interface (API). In some embodiments, the processor provides the resource consumption quantity to a billing service via an HTTP API.

**[0089]** In some aspects, a non-transitory computer readable storage medium includes instructions stored thereon that, when executed by a processor, cause the processor to receive, at a server, from a cluster of nodes on an edge network in communication with the server, a resource consumption data of a service hosted on the edge network; determine, based on a metering policy, a unit of measurement; and calculate a resource consumption quantity according to the unit of measurement.

**[0090]** In some aspects, the resource consumption data includes one or more data points, and each data point of the one or more data points includes a resource identifier, a time stamp, and a resource quantity. In some aspects, the resource consumption quantity is used to determine an amount to charge a user registered with the cluster of nodes.

**[0091]** In some aspects, the server is a first public cloud and the edge network is one or more of an on-premises data center, a distributed data center, or a second public cloud different from the first public cloud. In some aspects, the unit of measurement includes one or more of a time granularity or a type of resource.

**[0092]** In some aspects, the resource consumption data indicates resource consumption at a cluster level and the resource consumption quantity indicates resource consumption at a user level. In some aspects, instructions stored on the storage medium that, when executed by a processor, further cause the processor to determine the cluster of nodes by retrieving license information associated with a user registered with the cluster of nodes from a registration service via a hypertext transfer protocol (HTTP) application programing interface (API).

**[0093]** In some aspects, an apparatus includes a processor and a memory, wherein the memory includes programmed instructions that, when executed by the processor, cause the apparatus to receive, at a server, from a cluster of nodes on an edge network in communication with the server, a resource consumption data of a service hosted on the edge network; determine, based on a metering policy, a unit of measurement; and calculate a resource consumption quantity according to the unit of measurement.

**[0094]** In some aspects, the memory includes programmed instructions that, when executed by the processor, further cause the apparatus to determine the cluster of nodes by retrieving license information associated with a user registered with the cluster of nodes from a registration service via a hypertext transfer protocol (HTTP) application programing interface (API).

**[0095]** In some aspects, a computer-implemented method includes receiving, at a server, from a cluster of nodes on an edge network in communication with the server, a resource consumption data of a service hosted on the edge network; determining, based on a metering policy, a unit of measurement; and calculating a resource consumption quantity according to the unit of measurement.

**[0096]** In some aspects, the method further includes determining the cluster of nodes by retrieving license information associated with a user registered with the cluster of nodes from a registration service via a hypertext transfer protocol (HTTP) application programing interface (API).

**[0097]** Referring now to FIG. **6**, a flowchart of an example method **600** for collecting resource consumption data is illustrated, in accordance with some embodiments of the present disclosure. The method **600** can be performed by one or more systems, components, or modules depicted in FIGS. **1-4**, including, for example, the collector **130**, the aggregate collector **224**, the collector frame service **236**, etc. In some embodiments, instructions for performing the method **600** are executed by a processor included in, or associated with, the one or more systems, components, or modules and stored in a non-transitory computer readable storage medium

included in, or associated with, the one or more systems, components, or modules. Additional, fewer, or different operations may be performed in the method **600** depending on the embodiment. One or more operations of the method **600** can be combined with one or more operations of the method **500**.

**[0098]** According to the method **600**, a processor (e.g., the collector **130** or a processor therein) identifies, at an edge network (e.g., the edge network **110**), resource consumption data (e.g., service resource consumption data **222**, cluster resource consumption data **226**, buffered cluster resource consumption data **238**) (at operation **610**). The resource consumption data may be associated with a service, a cluster, a super-cluster, etc. In some embodiments, resource consumption data of one service can be combined with resource consumption data of another service (e.g., in one transmission packet or multi-part transmission) and provided together to the remote server. In some embodiments, the resource consumption data includes a status that indicates whether a service (e.g., the service **125**A) hosted on a cluster (e.g., the cluster **120**A) of nodes (e.g., the nodes **206**A-**206**K) on the edge network is powered on. In some embodiments, the resource consumption data includes one or more of a type of resource being consumed by the service, a quantity of the resource being consumed by the service, or a timestamp associated with the resource being consumed by the service. In some embodiments, the resource consumption data is collected, identified, and provided in accordance with a collector configuration (e.g., collected at a predetermined interval, granularity, etc.).

**[0099]** In some embodiments, the processor provides, to a remote server (e.g., the server **105**) in communication with the edge network, the resource consumption data (at operation **620**). In some embodiments, the processor receives an indication that communication with the remote server is interrupted. In some embodiments, the processor receives an indication that communication with the remote server is reestablished/restored. In some embodiments, the processor provides, in response to receiving the indication that communication is reestablished, the status, the type of resource, the quantity of the resource, and the resource consumption data.

**[0100]** In some embodiments, the remote server determines that the service is powered off in response to receiving an indication that communication with the edge network is active (e.g., the server can send a first health query to the edge network and can receive a failure code in response), and not receiving the resource consumption data for a predetermined time period. The server can determine compare a time difference between a most recent resource consumption data and a second most recent resource consumption data to determine if the time difference is greater than the predetermined time period.

**[0101]** In some aspects, a non-transitory computer readable storage medium includes instructions stored thereon that, when executed by a processor, cause the processor to identify, at an edge network, resource consumption data including a status that indicates whether a service hosted on a cluster of nodes on the edge network is powered on, a type of a resource being consumed by the service, a quantity of the resource being consumed by the service, and a time stamp associated with the resource being consumed by the service; and provide, to a remote server in communication with the edge network, the resource consumption data, wherein the remote server meters resource consumption based on the resource consumption data. In some aspects, the indication whether the service hosted on the cluster of nodes on the edge network is powered on includes a second indication of whether the edge network is in a dark-site mode.

**[0102]** In some aspects, instructions stored on the storage medium that, when executed by a processor, further cause the processor to receive an indication that communication with the remote server is interrupted. In some aspects, instructions stored on the storage medium that, when executed by a processor, further cause the processor to receive a second indication that communication with the remote server is restored; and provide, in response to receiving the second indication that communication is restored, the resource consumption data to the remote server.

**[0103]** In some aspects, the remote server determines that the service is powered off in response to: receiving an indication that communication with the edge network is active; and not receiving the resource consumption data for a predetermined time period.

**[0104]** In some aspects, instructions stored on the storage medium that, when executed by a processor, further cause the processor to combine the resource consumption data of the service hosted on the cluster of nodes with second resource consumption data of a second service external to the cluster of nodes. In some aspects, instructions stored on the storage medium that, when executed by a processor, further cause the processor to collect the resource consumption data periodically in accordance with a collector configuration.

**[0105]** In some aspects, an apparatus includes a processor and a memory, wherein the memory includes programmed instructions that, when executed by the processor, cause the apparatus to identify, at an edge network, resource consumption data including a status that indicates whether a service hosted on a cluster of nodes on the edge network is powered on, a type of a resource being consumed by the service, a quantity of the resource being consumed by the service, and a time stamp associated with the resource being consumed by the service; and provide, to a remote server in communication with the edge network, the resource consumption data, wherein the remote server meters resource consumption based on the resource consumption data.

**[0106]** In some aspects, the memory includes programmed instructions that, when executed by the processor, further cause the apparatus to receive an indication that communication with the remote server is interrupted. In some aspects, the memory includes programmed instructions that, when executed by the processor, further cause the apparatus to receive a second indication that communication with the remote server is restored; and provide, in response to receiving the second indication that communication is restored, the resource consumption data to the remote server.

**[0107]** In some aspects, the memory includes programmed instructions that, when executed by the processor, further cause the apparatus to combine the resource consumption data of the service hosted on the cluster of nodes with second resource consumption data of a second service external to the cluster of nodes. In some aspects, the memory includes programmed instructions stored thereon that, when executed by a processor, further cause the processor to collect the resource consumption data periodically in accordance with a collector configuration.

**[0108]** In some aspects, a computer-implemented method includes identifying, at an edge network, resource consumption data including a status that indicates whether a service hosted on a cluster of nodes on the edge network is powered on, a type of a resource being consumed by the service, a quantity of the resource being consumed by the service, and a time stamp associated with the resource being consumed by the service; and providing, to a remote server in communication with the edge network, the resource consumption data, wherein the remote server meters resource consumption based on the resource consumption data.

**[0109]** In some aspects, the method includes receiving an indication that communication with the remote server is interrupted. In some aspects, the method includes receiving a second indication that communication with the remote server is restored; and providing, in response to receiving the second indication that communication is restored, the resource consumption data to the remote server.

**[0110]** In some aspects, the method includes combining the resource consumption data of the service hosted on the cluster of nodes with second resource consumption data of a second service external to the cluster of nodes. In some aspects, the method includes collecting the resource consumption data periodically in accordance with a collector configuration.

**[0111]** Referring now to FIG. **7**, a flowchart of an example method **700** for updating resource consumption is illustrated, in accordance with some embodiments of the present disclosure. The method **700** can be performed by one or more systems, components, or modules depicted in FIGS. **1-4**, including, for example, the server **105**, the metering service **145**, the billing service **160**, etc. In some embodiments, instructions for performing the method **700** are executed by a processor included in, or associated with, the one or more systems, components, or modules and stored in a non-transitory computer readable storage medium included in, or associated with, the one or more systems, components, or modules. Additional, fewer, or different operations may be performed in the method **700** depending on the embodiment. One or more operations of the method **700** can be combined with one or more operations of one or more of the methods **500-600**.

**[0112]** According to the method **700**, a processor (e.g., the metering service **145** or a processor therein) receives, at a server (e.g., the server **105**), from a first cluster (e.g., the cluster **120**A) of nodes (e.g., the nodes **206**A-**206**K) on an edge network (e.g., the edge network **110**) in communication with the server, first resource consumption data (e.g., the buffered cluster resource consumption data **238**, the detailed metering item **142**, etc.) of a service (e.g., the service **125**A) hosted on the edge network (at operation **710**). In some embodiments, the first resource consumption data is collected at a first time. In some embodiments, the first cluster is registered (e.g., by the registration service **115**) to a user under a consumption-based license model in which the user is to pay based on a quantity of resources consumed by the first cluster and other clusters registered to the user.

**[0113]** In some embodiments, the processor calculates a first resource consumption quantity (e.g., a charge item **148**) based on the first resource consumption data (at operation **720**). In some embodiments, the processor sends the first resource consumption quantity to a billing service (e.g., the billing service **160**). In some embodiments, the billing service one of overcharges or undercharges a user registered to the first cluster of nodes and the second cluster of nodes.

**[0114]** In some embodiments, the processor receives, from the a second cluster of nodes (or a node in the first cluster of nodes) on the edge network, delayed resource consumption (e.g., another instance of the buffered cluster resource consumption data **238**, another instance of the detailed metering item **142**, etc.) data that is collected at the first time (at operation **730**). In some embodiments, at least a part of the delayed resource consumption data was not available to be received when the first resource consumption data was received (e.g., due to a source failure of the second cluster of nodes or a node of the first cluster of nodes, a network failure, or the second cluster of nodes or a node of the first cluster of nodes operating in dark-site mode). In some embodiments, the delayed resource consumption data includes the first resource consumption data (e.g., the resource consumption data of the first cluster of nodes that were available to be received when the first resource consumption data was received). In some embodiments, the delayed resource consumption data only includes resource consumption data that was not available to be received when the first resource consumption data was received. In some embodiments, the second cluster is registered to the user under the consumption-based license model.

**[0115]** In some embodiments, the processor calculates a second resource consumption quantity based on the delayed resource consumption data (at operation **740**). In some embodiments, the processor sends the second resource consumption quantity to a billing service (e.g., the billing service **160**). In some embodiments, the processor or the billing service compares the first resource consumption quantity to the second resource consumption quantity to determine that the second resource consumption quantity is different than the first resource consumption quantity. In some embodiments, the second resource consumption quantity in response to determining that the second resource consumption quantity being different than the first resource consumption quantity.

**[0116]** In some embodiments, the billing service replaces the first resource consumption quantity with the second resource consumption quantity, or otherwise updates the first resource consumption quantity to include the second resource consumption quantity. In response, the billing service does the replacement/update in response to determining that the second resource consumption quantity being different than the first resource consumption quantity. In some embodiments, the second resource consumption quantity includes the resources consumed by the first cluster at the first time (e.g., the first resource consumption quantity and additional resource consumed by the first cluster at the first time). In some embodiments, the billing service provides (e.g., presents, displays), to the user, the first resource consumption quantity and the second resource consumption quantity. In some embodiments, in response to the billing service receiving the second resource consumption quantity, the billing service charges a user registered to the first cluster of nodes and the second cluster of nodes a correct amount based on the resources used by the user and the resource consumption license model for the user.

**[0117]** In some embodiments, the processor pre-configures a time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity. In some embodiments, the processor

adjusts the preconfigured time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity in response to a user request.

**[0118]** In some aspects, a non-transitory computer readable storage medium includes instructions stored thereon that, when executed by a processor, cause the processor to receive, at a server, from a first cluster of nodes on an edge network in communication with the server, first resource consumption data of a first service hosted on the edge network, wherein the first resource consumption data is collected at a first time; calculate a first resource consumption quantity based on the first resource consumption data; receive, from a second cluster of nodes on the edge network, delayed resource consumption data of a second service hosted on the edge network, wherein the delayed resource consumption data collected at the first time; and calculate a second resource consumption quantity based on the delayed resource consumption data.

**[0119]** In some aspects, the medium includes instructions stored thereon that, when executed by a processor, further cause the processor to send the first resource consumption quantity to a billing service; and send the second resource consumption quantity to the billing service, wherein the billing service updates the first resource consumption quantity to include the second resource consumption quantity. In some aspects, the medium includes instructions stored thereon that, when executed by a processor, further cause the processor to determine that the second resource consumption quantity is different than the first resource consumption quantity; and send the second resource consumption quantity in response to determining that the second resource consumption quantity being different than the first resource consumption quantity.

**[0120]** In some aspects, the medium includes instructions stored thereon that, when executed by a processor, further cause the processor to preconfigure a time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity. In some aspects, the medium includes instructions stored thereon that, when executed by a processor, further cause the processor to adjust the preconfigured time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity in response to a user request.

**[0121]** In some aspects, the delayed resource consumption data was not available to be received at a same time as the first resource consumption data due to an outage. In some aspects, the outage is one of a source failure of the second cluster of nodes, a network failure of a communication network between the server and the edge network, or the second cluster of nodes operating as a dark-site.

**[0122]** In some aspects, an apparatus includes a processor and a memory. In some embodiments, the memory includes programmed instructions that, when executed by the processor, cause the apparatus to receive, at a server, from a first cluster of nodes on an edge network in communication with the server, first resource consumption data of a first service hosted on the edge network, wherein the first resource consumption data is collected at a first time; calculate a first resource consumption quantity based on the first resource consumption data; receive, from a second cluster of nodes on the edge network, delayed resource consumption data of a second service hosted on the edge network, wherein the delayed resource consumption data collected at the first time; and calculate a second resource consumption quantity based on the delayed resource consumption data.

**[0123]** In some aspects, the memory includes programmed instructions that, when executed by the processor, further cause the apparatus to: send the first resource consumption quantity to a billing service; and send the second resource consumption quantity to the billing service, wherein the billing service updates the first resource consumption quantity to include the second resource consumption quantity. In some aspects, the memory includes programmed instructions that, when executed by the processor, further cause the apparatus to: determine that the second resource consumption quantity is different than the first resource consumption quantity; and send the second resource consumption quantity in response to determining that the second resource consumption quantity being different than the first resource consumption quantity.

**[0124]** In some aspects, the memory includes programmed instructions that, when executed by the processor, further cause the apparatus to preconfigure a time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity. In some aspects, the memory includes programmed instructions that, when executed by the processor, further cause the apparatus to adjust the preconfigured time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity in response to a user request.

**[0125]** In some aspects, a computer-implemented method includes receiving, at a server, from a first cluster of nodes on an edge network in communication with the server, first resource consumption data of a first service hosted on the edge network, wherein the first resource consumption data is collected at a first time; calculating a first resource consumption quantity based on the first resource consumption data; receiving, from a second cluster of nodes on the edge network, delayed resource consumption data of a second service hosted on the edge network, wherein the delayed resource consumption data collected at the first time; and calculating a second resource consumption quantity based on the delayed resource consumption data.

**[0126]** In some aspects, the method includes sending the first resource consumption quantity to a billing service; and sending the second resource consumption quantity to the billing service, wherein the billing service updates the first resource consumption quantity to include the second resource consumption quantity. In some aspects, the method includes determining that the second resource consumption quantity is different than the first resource consumption quantity; and sending the second resource consumption quantity in response to determining that the second resource consumption quantity being different than the first resource consumption quantity.

**[0127]** In some aspects, the method includes preconfiguring a time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity. In some aspects, the method includes adjusting the preconfigured time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity in response to a user request.

**[0128]** Referring now to FIG. **8**, a flowchart of an example method **800** for providing alerts is illustrated, in accordance

with some embodiments of the present disclosure. The method **800** can be performed by one or more systems, components, or modules depicted in FIGS. **1-4**, including, for example, the server **105**, the metering service **145**, the alerts service **155**, etc. In some embodiments, instructions for performing the method **800** are executed by a processor included in, or associated with, the one or more systems, components, or modules and stored in a non-transitory computer readable storage medium included in, or associated with, the one or more systems, components, or modules. Additional, fewer, or different operations may be performed in the method **800** depending on the embodiment. One or more operations of the method **800** can be combined with one or more operations of one or more of the methods **500-700**.

**[0129]** According to the method **800**, a processor (e.g., the alerts service **155** or a processor therein) determines an issue (at operation **810**). In some embodiments, the issue includes not receiving resource consumption data (e.g., the buffered cluster resource consumption data **238**) before a task (e.g., the task **325**, a regular task, a fixer task) is executed. In some embodiments, the resource consumption data is collected in a cluster (e.g., the cluster **120**A) on an edge network (e.g., the edge network **110**) at a first time. In some embodiments, the task corresponds to the first time (e.g., the task includes other data collected at the first time). In some embodiments, the task is executed in a server (e.g., the server **105**) coupled to the edge network. In some embodiments, the processor determines that the resource consumption data collected at the first time is received within a predetermined time after the task (e.g., data delay). In some embodiments, the processor determines that the resource consumption data collected at the first time is not received within a predetermined time after the task (e.g., data loss).

**[0130]** In some embodiments, the issue includes not connecting to either a first application programming interface (API) for registering the cluster or a second API for providing a charge item corresponding to the resource consumption data. In some embodiments, the processor alerts a user or a site reliability engineer (SRE) of the issue (at operation **820**).

**[0131]** In some aspects, a non-transitory computer readable storage medium includes instructions stored thereon that, when executed by a processor, cause the processor to determine an issue and alert a user or a site reliability engineer (SRE) of the issue. In some aspects, the issue includes one or more of not receiving resource consumption data before a task is executed or not connecting to either an first application programming interface (API) for registering the cluster or a second API for providing a charge item corresponding to the resource consumption data. In some aspects, the resource consumption data is collected in a cluster on an edge network at a first time. In some aspects, the task includes other data collected at the first time. In some aspects, the task is executed in a server coupled to the edge network.

**[0132]** In some aspects, the medium includes instructions stored thereon that, when executed by a processor, further cause the processor to determine that the resource consumption data collected at the first time is received within a predetermined time after the task. In some aspects, the medium includes instructions stored thereon that, when executed by a processor, further cause the processor to determine that the resource consumption data collected at the first time is not received within a predetermined time after the task.

**[0133]** Referring now to FIG. **9**, a flowchart of an example method **900** for validating a metering system is illustrated, in accordance with some embodiments of the present disclosure. The method **900** can be performed by one or more systems, components, or modules depicted in FIGS. **1-4**, including, for example, the server **105**, the validation service **405**, etc. In some embodiments, instructions for performing the method **900** are executed by a processor included in, or associated with, the one or more systems, components, or modules and stored in a non-transitory computer readable storage medium included in, or associated with, the one or more systems, components, or modules. Additional, fewer, or different operations may be performed in the method **900** depending on the embodiment. One or more operations of the method **900** can be combined with one or more operations of one or more of the methods **500-800**.

**[0134]** According to the method **900**, a processor (e.g., the validation service **405** or a processor therein) provides input data to a cluster or a first service related to metering resource consumption of the cluster under a consumption-based license model (at operation **910**). In some embodiments, the input data is a cluster configuration provided to the first service and the first service is the registration service **115**. In some embodiments, the input data is a workload provided to the cluster (e.g., one or more services being metered that are a part of the cluster).

**[0135]** In some embodiments, the processor queries the first service or a second service related to metering the resource consumption of the cluster under the consumption-based license model (at operation **920**). In some embodiments, the service being queried is the first service (e.g., the registration service **115**) and the query is whether the cluster is registered. In some embodiments, the service being queried is the second service (e.g., one of the data processing pipeline **135**, the metering service **145**, or the billing service **160**) and the query is an amount/quantity of resources consumed.

**[0136]** In some embodiments, the processor receives an actual response from the first service or the second service based on the query (at operation **930**). In some embodiments, the processor compares the actual response to an expected response (at operation **940**). In some embodiments, the processor determines whether the actual response matches the expected response (at operation **950**).

**[0137]** In some aspects, a non-transitory computer readable storage medium includes instructions stored thereon that, when executed by a processor, cause the processor to provide input data to a cluster or a first service related to metering resource consumption of the cluster under a consumption-based license model, query one of the first service or a second service related to metering the resource consumption of the cluster under the consumption-based license model, receive an actual response from the one of the first service or the second service based on the query, compare the actual response to an expected response, and determine whether the actual response matches the expected response.

**[0138]** In some aspects, the input data is a cluster configuration, the first service is the registration service, and the query is whether the cluster is registered. In some aspects, the input data is a workload, the second service is one of the

data processing pipeline, the metering service, or the billing service, and the query is an amount of resources consumed.

**[0139]** Referring now to FIG. **10**, a flowchart of an example method **1000** for registering a cluster under the consumption-based license model is illustrated, in accordance with some embodiments of the present disclosure. The method **1000** can be performed by one or more systems, components, or modules depicted in FIGS. **1-4**, including, for example, the server **105**, the registration service **115**, etc. In some embodiments, instructions for performing the method **1000** are executed by a processor included in, or associated with, the one or more systems, components, or modules and stored in a non-transitory computer readable storage medium included in, or associated with, the one or more systems, components, or modules. Additional, fewer, or different operations may be performed in the method **1000** depending on the embodiment. One or more operations of the method **1000** can be combined with one or more operations of one or more of the methods **500-900**.

**[0140]** According to the method **1000**, a processor (e.g., the registration service **115** or a processor therein) receives a registration request from a user to register a cluster (e.g., the cluster **120**A) or a super-cluster (e.g., the super-cluster **240**) under a consumption-based license (at operation **1010**). In some embodiments, the user is a service provider. In some embodiments, the processor generates an application programming interface (API) key, or other token, for the user to consume resources on the cluster or super-cluster based on (e.g., according to) the consumption-based license (at operation **1020**). In some embodiments, the cluster or super-cluster is on an edge network (e.g., the edge network **110**).

**[0141]** In some embodiments, the processor determines whether the cluster or super-cluster is under a term-based license (at operation **1030**). In some embodiments, in response to the processor determining that the cluster or super-cluster is under the term-based license, the processor revokes the term-based license (at operation **1040**). In some embodiments, the processor transfers credits from the term-based license to the consumption-based license.

**[0142]** In some embodiments, the processor assigns the API key to the cluster or super-cluster (at operation **1050**). In some embodiments, the cluster or super-cluster stores the API key locally to apply the consumption-based license. In some embodiments, if the cluster or super-cluster has another API key for the term-based license, the cluster or super-cluster deletes the other API key over overwrites the other API key with the API key. In some embodiments, if the super-cluster stores the API key locally, then the consumption-based license applies to all clusters of the super-cluster.

**[0143]** In some embodiments, the processor receives a registration request from a user to register one or more services on a cluster under a consumption-based license. In some embodiments, the processor registers one or more other services under the term-based license or the one or more other services are already registered under the term-based license. In some embodiments, upon the API key being stored in the cluster, the consumption-based license is only applied to the one or more services and the term-based license remains applied to the one or more other services.

**[0144]** In some aspects, a non-transitory computer readable storage medium comprising instructions stored thereon that, when executed by a processor, cause the processor to receive a registration request from a user, generate an application programming interface (API) key for the user to consume resources in a cluster based on a consumption-based license, and assign the API key to the cluster. In some aspects, the cluster is on an edge network. In some aspects, the cluster stores the API key locally to apply the consumption-based license.

**[0145]** In some aspects, the medium includes instructions stored thereon that, when executed by a processor, further cause the processor to determine whether the cluster is under a term-based license, and, in response to determining that the cluster is under the term-based license, revoke the term-based license. In some aspects, the medium includes instructions stored thereon that, when executed by a processor, cause the processor to further transfer credits from the term-based license to the consumption based license. In some aspects, the user is a service provider.

**[0146]** Each of the components/elements/entities (e.g., the server **105**, the edge network **110**, the registration service **115**, the cluster **120**A, the collector **130**, the data processing pipeline **135**, the data repository **140**, the metering service **145**, the metering storage **150**, the alerts service **155**, the billing service **160**, the consumption collector **220**, the aggregate collector **224**, the cluster repository **228**, the collector frame service **236**, the metering master **305**, the metering worker **310**A, the validation service **405**, etc.) of the computing environments (e.g., the computing environment **100**, the computing environment **300**, the computing environment **400**), is implemented using hardware, software, or a combination of hardware or software, in one or more embodiments. One or more of the components of the computing environments may include a processor with instructions or may be an apparatus/device (e.g., server) including a processor with instructions, in some embodiments. In some embodiments, multiple components may be part of a same apparatus and/or share a same processor. Each of the components of the computing environments can include any application, program, library, script, task, service, process or any type and form of executable instructions executed by one or more processors, in one or more embodiments. Each of the one or more processors is hardware, in some embodiments. The instructions may be stored on one or more computer readable and/or executable storage media including non-transitory storage media.

**[0147]** The herein described subject matter sometimes illustrates different components contained within, or connected with, different other components. It is to be understood that such depicted architectures are merely exemplary, and that in fact many other architectures can be implemented which achieve the same functionality. In a conceptual sense, any arrangement of components to achieve the same functionality is effectively "associated" such that the desired functionality is achieved. Hence, any two components herein combined to achieve a particular functionality can be seen as "associated with" each other such that the desired functionality is achieved, irrespective of architectures or intermedial components. Likewise, any two components so associated can also be viewed as being "operably connected," or "operably coupled," to each other to achieve the desired functionality, and any two components capable of being so associated can also be viewed as being "operably couplable," to each other to achieve the desired functionality. Specific examples of operably couplable include but are not limited to physically mateable and/or physically interacting components and/or wirelessly interactable and/or

wirelessly interacting components and/or logically interacting and/or logically interactable components.

**[0148]** With respect to the use of substantially any plural and/or singular terms herein, those having skill in the art can translate from the plural to the singular and/or from the singular to the plural as is appropriate to the context and/or application. The various singular/plural permutations may be expressly set forth herein for sake of clarity.

**[0149]** It will be understood by those within the art that, in general, terms used herein, and especially in the appended claims (e.g., bodies of the appended claims) are generally intended as "open" terms (e.g., the term "including" should be interpreted as "including but not limited to," the term "having" should be interpreted as "having at least," the term "includes" should be interpreted as "includes but is not limited to," etc.). It will be further understood by those within the art that if a specific number of an introduced claim recitation is intended, such an intent will be explicitly recited in the claim, and in the absence of such recitation no such intent is present. For example, as an aid to understanding, the following appended claims may contain usage of the introductory phrases "at least one" and "one or more" to introduce claim recitations. However, the use of such phrases should not be construed to imply that the introduction of a claim recitation by the indefinite articles "a" or "an" limits any particular claim containing such introduced claim recitation to disclosures containing only one such recitation, even when the same claim includes the introductory phrases "one or more" or "at least one" and indefinite articles such as "a" or "an" (e.g., "a" and/or "an" should typically be interpreted to mean "at least one" or "one or more"); the same holds true for the use of definite articles used to introduce claim recitations. In addition, even if a specific number of an introduced claim recitation is explicitly recited, those skilled in the art will recognize that such recitation should typically be interpreted to mean at least the recited number (e.g., the bare recitation of "two recitations," without other modifiers, typically means at least two recitations, or two or more recitations). Furthermore, in those instances where a convention analogous to "at least one of A, B, and C, etc." is used, in general such a construction is intended in the sense one having skill in the art would understand the convention (e.g., "a system having at least one of A, B, and C" would include but not be limited to systems that have A alone, B alone, C alone, A and B together, A and C together, B and C together, and/or A, B, and C together, etc.). In those instances where a convention analogous to "at least one of A, B, or C, etc." is used, in general such a construction is intended in the sense one having skill in the art would understand the convention (e.g., "a system having at least one of A, B, or C" would include but not be limited to systems that have A alone, B alone, C alone, A and B together, A and C together, B and C together, and/or A, B, and C together, etc.). It will be further understood by those within the art that virtually any disjunctive word and/or phrase presenting two or more alternative terms, whether in the description, claims, or drawings, should be understood to contemplate the possibilities of including one of the terms, either of the terms, or both terms. For example, the phrase "A or B" will be understood to include the possibilities of "A" or "B" or "A and B." Further, unless otherwise noted, the use of the words "approximate," "about," "around," "substantially," etc., mean plus or minus ten percent.

**[0150]** The foregoing description of illustrative embodiments has been presented for purposes of illustration and of description. It is not intended to be exhaustive or limiting with respect to the precise form disclosed, and modifications and variations are possible in light of the above teachings or may be acquired from practice of the disclosed embodiments. It is intended that the scope of the disclosure be defined by the claims appended hereto and their equivalents.

**1**. A non-transitory computer readable storage medium comprising instructions stored thereon that, when executed by a processor, cause the processor to:

receive, at a server, from a first cluster of nodes on a network in communication with the server, first resource consumption data of a first service hosted on the network, wherein the first resource consumption data is collected at a first time;

calculate a first resource consumption quantity based on the first resource consumption data;

receive, from a second cluster of nodes on the network, second resource consumption data of a second service hosted on the network, wherein the second resource consumption data is collected at the first time but was unavailable at the server for calculation of the first resource consumption quantity; and

calculate a second resource consumption quantity based on the first resource consumption data and the second resource consumption data.

**2**. The storage medium of claim **1**, comprising instructions stored thereon that, when executed by a processor, further cause the processor to:

send the first resource consumption quantity to a billing service; and

send the second resource consumption quantity to the billing service, wherein the billing service updates the first resource consumption quantity to include the second resource consumption quantity.

**3**. The storage medium of claim **1**, comprising instructions stored thereon that, when executed by a processor, further cause the processor to:

determine that the second resource consumption quantity is different than the first resource consumption quantity; and

send the second resource consumption quantity in response to determining that the second resource consumption quantity being different than the first resource consumption quantity.

**4**. The storage medium of claim **1**, comprising instructions stored thereon that, when executed by a processor, further cause the processor to preconfigure a time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity.

**5**. The storage medium of claim **4**, comprising instructions stored thereon that, when executed by a processor, further cause the processor to adjust the preconfigured time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity in response to a user request.

**6**. The storage medium of claim **1**, wherein the second resource consumption data was not available to be received at a same time as the first resource consumption data due to an outage.

**7**. The storage medium of claim **6**, wherein the outage is one of a source failure of the second cluster of nodes, a

network failure of a communication network between the server and the network, or the second cluster of nodes operating as a dark-site.

**8**. An apparatus comprising a processor and a memory, wherein the memory includes programmed instructions that, when executed by the processor, cause the apparatus to:

receive, at a server, from a first cluster of nodes on a network in communication with the server, first resource consumption data of a first service hosted on the network, wherein the first resource consumption data is collected at a first time;

calculate a first resource consumption quantity based on the first resource consumption data;

receive, from a second cluster of nodes on the network, second resource consumption data of a second service hosted on the network, wherein the second resource consumption data is collected at the first time but was unavailable at the server for calculation of the first resource consumption quantity; and

calculate a second resource consumption quantity based on the first resource consumption data and the second resource consumption data.

**9**. The apparatus of claim **8**, wherein the memory includes programmed instructions that, when executed by the processor, further cause the apparatus to:

send the first resource consumption quantity to a billing service; and

send the second resource consumption quantity to the billing service, wherein the billing service updates the first resource consumption quantity to include the second resource consumption quantity.

**10**. The apparatus of claim **8**, wherein the memory includes programmed instructions that, when executed by the processor, further cause the apparatus to:

determine that the second resource consumption quantity is different than the first resource consumption quantity; and

send the second resource consumption quantity in response to determining that the second resource consumption quantity being different than the first resource consumption quantity.

**11**. The apparatus of claim **8**, wherein the memory includes programmed instructions that, when executed by the processor, further cause the apparatus to preconfigure a time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity.

**12**. The apparatus of claim **11**, wherein the memory includes programmed instructions that, when executed by the processor, further cause the apparatus to adjust the preconfigured time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity in response to a user request.

**13**. The apparatus of claim **8**, wherein the second resource consumption data was not available to be received at a same time as the first resource consumption data due to an outage.

**14**. The apparatus of claim **13**, wherein the outage is one of a source failure of the second cluster of nodes, a network failure of a communication network between the server and the network, or the second cluster of nodes operating as a dark-site.

**15**. A computer-implemented method comprising:

receiving, at a server, from a first cluster of nodes on a network in communication with the server, first resource consumption data of a first service hosted on the network, wherein the first resource consumption data is collected at a first time;

calculating a first resource consumption quantity based on the first resource consumption data;

receiving, from a second cluster of nodes on the network, second resource consumption data of a second service hosted on the network, wherein the second resource consumption data is collected at the first time but was unavailable at the server for calculation of the first resource consumption quantity; and

calculating a second resource consumption quantity based on the first resource consumption data and the second resource consumption data.

**16**. The method of claim **15**, further comprising:

sending the first resource consumption quantity to a billing service; and

sending the second resource consumption quantity to the billing service, wherein the billing service updates the first resource consumption quantity to include the second resource consumption quantity.

**17**. The method of claim **15**, further comprising:

determining that the second resource consumption quantity is different than the first resource consumption quantity; and

sending the second resource consumption quantity in response to determining that the second resource consumption quantity being different than the first resource consumption quantity.

**18**. The method of claim **15**, further comprising preconfiguring a time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity.

**19**. The method of claim **18**, further comprising adjusting the preconfigured time delta between calculating the first resource consumption quantity and calculating the second resource consumption quantity in response to a user request.

**20**. The method of claim **15**, wherein the second resource consumption data was not available to be received at a same time as the first resource consumption data due to an outage.

**21**. The storage medium of claim **1**, wherein the second resource consumption data was not available to be received at a same time as the first resource consumption data due to the second cluster of nodes operating as a dark-site.

**22**. The apparatus of claim **8**, wherein the second resource consumption data was not available to be received at a same time as the first resource consumption data due to the second cluster of nodes operating as a dark-site.

**23**. The method of claim **15**, wherein the second resource consumption data was not available to be received at a same time as the first resource consumption data due to the second cluster of nodes operating as a dark-site.

* * * * *