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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 January 2025 (23.01.2025)

WIPO | PCT

(10) International Publication Number
WO 2025/017677 A1

(51) International Patent Classification:
*H04L 43/08* (2022.01) *H04L 41/046* (2022.01)
*H04L 67/10* (2022.01)

(21) International Application Number:
PCT/IN2024/051238

(22) International Filing Date:
15 July 2024 (15.07.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
202321047842 15 July 2023 (15.07.2023) IN

(71) Applicant: **JIO PLATFORMS LIMITED** [IN/IN]; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN).

(72) Inventors: **BHATNAGAR, Aayush**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN). **MURARKA, Ankit**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN). **KOLARIYA, Jugal kishore**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN). **KUMAR, Gaurav**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN). **SAHU, Kishan**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN). **VERMA, Rahul**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN). **MEENA, Sunil**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN). **GURBANI, Gourav**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN). **CHAUDHARY, Sanjana**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN). **GANVEER, Chandra kumar**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN). **DE, Supriya**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN). **DEBASHISH, Kumar**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMED-

(54) Title: SYSTEM AND METHOD FOR MONITORING INFRASTRUCTURE IN A CLOUD NETWORK

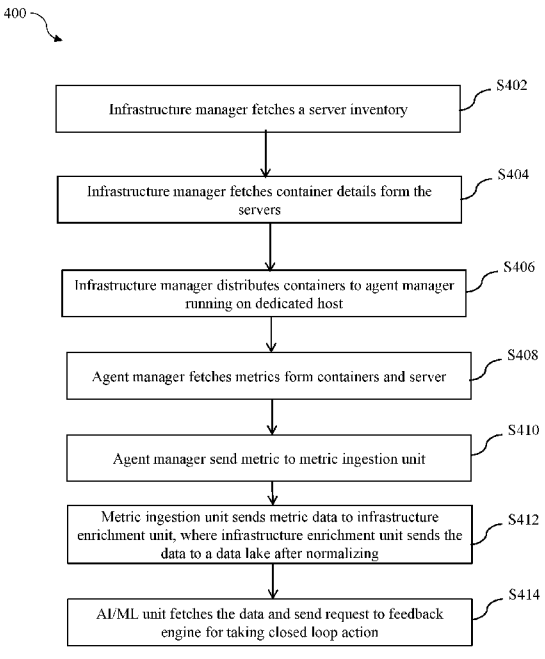


FIG. 4

(57) Abstract: A system(125) and a method for monitoring infrastructure in a cloud network(110) is described The method includes capturing a server inventory from the cloud network(110). A plurality of container details is extracted from the captured server inventory. An Internet Protocol (IP) address is assigned to one or more agent managers, and the plurality of container details is sent to the one or more agent mangers. The plurality of container details and server details are processed at the one or more agent managers to obtain a metric data.

WO 2025/017677 A1

[Continued on next page]

ABAD 380006 (IN). **MEHUL, Tilala**; OFFICE-101, SAFFRON, NR. CENTRE POINT, PANCHWATI 5 RASTA, AMBAWADI, GUJARAT, AHMEDABAD 380006 (IN).

(74) **Agent: JAPHET, Chinthan**; K Law (Krishnamurthy and Co.), 4th Floor, Prestige Takt, No 23, Kasturba Road Cross, Bangalore, Bangalore 560 001 (IN).

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

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

**Declarations under Rule 4.17:**

— *of inventorship (Rule 4.17(iv))*

**Published:**

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

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

# SYSTEM AND METHOD FOR MONITORING INFRASTRUCTURE IN A CLOUD NETWORK

## FIELD OF THE INVENTION

**[0001]** The present subject matter relates to the field of cloud resource monitoring systems and methods. More particularly, the invention pertains to a system and a method for ensuring seamless cloud infrastructure monitoring in a cloud environment.

## BACKGROUND OF THE INVENTION

**[0002]** In existing legacy metrics methods and legacy metrics systems, a lot of manual processes need to be performed for running agents and collecting metrics. In cloud environments, agents refer to software components or modules that perform specific tasks or functions within the cloud infrastructure. These agents are typically deployed on virtual machines (VMs), containers, or directly on cloud services to facilitate various operational, management, security, or monitoring tasks. The metrics are used for monitoring and managing performance of resources of the cloud infrastructure, such as uptime, error rate, compute cost, and requests per minute. There does not exist a completely automated process for deploying a system for monitoring resources and operations in a cloud environment. Further, the monitoring is generally initiated with agents deployed on those servers which at times hinder performance of the applications running in the system. Hence, there is a need for a system and a method for ensuring seamless cloud infrastructure monitoring in a cloud environment in an effective manner.

## SUMMARY OF THE INVENTION

**[0003]** One or more embodiments of the present disclosure provide a system and method for monitoring infrastructure in the cloud network.

**[0004]** In one aspect of the present invention, a system for monitoring infrastructure in a cloud network is disclosed. The system includes a capturing unit configured to capture a server inventory from a cloud. The system further includes an extraction unit configured to extract a plurality of container details from the captured server inventory. The system further includes an assigning unit configured to assign an Internet Protocol (IP) address to one or more agent managers. The system further includes a transceiver configured to send the plurality of container details to the one or more agent managers. The system further includes a processing unit configured to process the plurality of container details and server details at the one or more agent managers to obtain metric data.

**[0005]** In one aspect, the one or more agent managers are hosted on at least one dedicated host. The assigned or allocated IP address corresponds to a host IP address associated with the at least one dedicated host. An updating unit is configured to add or remove host, to or from the at least one dedicated host. The processing unit is configured to process the plurality of container details and the server details by ingesting the metric data at the one or more agent managers from a broker. The ingesting comprises consuming, by the one or more agent managers, the metric data from the broker. Consuming the metric data from the broker enables obtaining a file in a pre-defined format. The file comprises validated metrics from the metric data. A fetching unit is configured to fetch the file from the broker with the pre-defined format comprising the validated metrics. The processing unit is configured to shrink the metric data received from an infrastructure enrichment unit, using data normalization, and store the metric data into a data lake. A detection unit is configured to detect anomalies in the validated metrics stored in the data lake. A triggering unit is configured to trigger a feedback unit upon detecting anomalies in the validated metrics stored in the data lake. Triggering of the feedback unit enables reporting and alarm engine to take up pre-emptive action.

**[0006]** In another aspect of the present invention, a method for monitoring infrastructure in a cloud network is disclosed. The method includes the step of

capturing a server inventory from the cloud network. The method further includes the step of capturing a server inventory from the cloud network. The method further includes the step of extracting a plurality of container details from the captured server inventory. The method further includes the step of assigning an Internet Protocol (IP) address to one or more agent managers. The method further includes the step of sending the plurality of container details to the one or more agent mangers. The method further includes the step of processing the plurality of container details and server details at the one or more agent managers to obtain a metric data.

**[0007]** In one aspect, the one or more agent managers are hosted on at least one dedicated host. The assigned or allocated IP address corresponds to a host IP address associated with the at least one dedicated host. The method further comprises adding or removing the host to or from the at least one dedicated host. Processing of the plurality of container details and the server details comprises the step of ingesting, the metric data by the one or more agent managers from a broker. The ingesting comprises consuming the metric data from the broker. Consuming the metric data from the broker enables obtaining a file in a pre-defined format. The file comprises validated metrics from the metric data. The method further comprises fetching the file from the broker with the pre-defined format comprising the validated metrics. The method further comprises shrinking the metric data received from an infrastructure enrichment unit, using data normalization, and store the metric data into a data lake. The method further comprises detecting anomalies in the validated metrics stored in the data lake. The method further comprises triggering a feedback unit upon detecting anomalies in the validated metrics stored in the data lake. Triggering of the feedback unit enables reporting and alarm engine to take pre-emptive action

**[0008]** Other features and aspects of this invention will be apparent from the following description and the accompanying drawings. The features and advantages described in this summary and in the following detailed description are not all-inclusive, and particularly, many additional features and advantages will be apparent to one of ordinary skill in the relevant art, in view of the drawings, specification, and

claims hereof. Moreover, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes and may not have been selected to delineate or circumscribe the inventive subject matter, resort to the claims being necessary to determine such inventive subject matter.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0009]** The accompanying drawings, which are incorporated herein, and constitute a part of this disclosure, illustrate exemplary embodiments of the disclosed methods and systems in which like reference numerals refer to the same parts throughout the different drawings. Components in the drawings are not necessarily to scale, emphasis instead being placed upon clearly illustrating the principles of the present disclosure. Some drawings may indicate the components using block diagrams and may not represent the internal circuitry of each component. It will be appreciated by those skilled in the art that disclosure of such drawings includes disclosure of electrical components, electronic components or circuitry commonly used to implement such components.

**[0010]** **FIG. 1** illustrates an environment including a system for monitoring infrastructure in a cloud network, according to one or more embodiments of the present disclosure;

**[0011]** **FIG. 2** illustrates a block diagram of the system for monitoring infrastructure in the cloud network, according to various embodiments of the present system;

**[0012]** **FIG. 3** illustrates a block diagram of the environment including the system for monitoring infrastructure in the cloud network, according to various embodiments of the present system; and

**[0013]** **FIG. 4** illustrates a flow chart of a method for monitoring infrastructure in the cloud network, according to one or more embodiments of the present disclosure.

**[0014]** The foregoing shall be more apparent from the following detailed description of the invention.

## DETAILED DESCRIPTION OF THE INVENTION

**[0015]** Some embodiments of the present disclosure, illustrating all its features, will now be discussed in detail. It must also be noted that as used herein and in the appended claims, the singular forms "a", "an" and "the" include plural references unless the context clearly dictates otherwise.

**[0016]** Various modifications to the embodiment will be readily apparent to those skilled in the art and the generic principles herein may be applied to other embodiments. However, one of ordinary skill in the art will readily recognize that the present disclosure including the definitions listed here below are not intended to be limited to the embodiments illustrated but is to be accorded the widest scope consistent with the principles and features described herein.

**[0017]** A person of ordinary skill in the art will readily ascertain that the illustrated steps detailed in the figures and here below are set out to explain the exemplary embodiments shown, and it should be anticipated that ongoing technological development will change the manner in which particular functions are performed. These examples are presented herein for purposes of illustration, and not limitation. Further, the boundaries of the functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternative boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed. Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the disclosed embodiments.

**[0018]** In a Cloud-Native Infrastructure Stack (CNIS) architecture, a proposed method can be used to perform multiple actions once an agent manager fetches a data from hosts. By using the proposed method, the CNIS architecture incorporates an agentless unique architectural design for efficiently fetching container and host server

metrics in a cloud environment. The CNIS works cohesively with various systems in a wireless network to ensure seamless cloud infrastructure monitoring with detailed and timely insights, so as to make a cloud observable in the CNIS architecture.

**[0019]** **FIG. 1** illustrates an environment **100** including a system **115** for monitoring infrastructure in a cloud network **110**. The environment **100** includes a plurality of hosts **105** (represented as a first host **105-1**, a second host **105-2**, and nth host **105-n**) connected with the cloud network **110**. The hosts **105** can be understood as computing devices communicating with each other in a computer network. The cloud network **110** can be understood as a Wide Area Network (WAN) hosting users and resources and allowing the two to communicate via cloud-based technologies. The cloud network **110** may consist of server, memories, virtual routers, firewalls, and network management software. The cloud network **110** may be a public, private, hybrid, or a community cloud network.

**[0020]** The environment **100** further includes the system **115** for managing servers configured in the cloud network **110**. The system **115** is communicably coupled with the plurality of hosts **105** and the cloud network **110**. The system **115** may be a computing device having a User Interface (UI), such as a laptop, general-purpose computer, desktop, personal digital assistant, tablet computer, mainframe computer.

**[0021]** In some implementations, the system **115** may include by way of example but not limitation, one or more of a standalone server, a server blade, a server rack, a bank of servers, a server farm, hardware supporting a part of a cloud service or system, a home server, hardware running a virtualized server, one or more processors executing code to function as a server, one or more machines performing server-side functionality as described herein, at least a portion of any of the above, some combination thereof.

**[0022]** The environment **100** further includes a data lake **120** communicably coupled to the system **115**. The data lake **120** is a data repository providing storage for structured and unstructured data, such as for machine learning, streaming, or data

science. The data lake **120** allows users and/or organizations to ingest and manage large volumes of data in an aggregated storage solution for business intelligence or data products.

**[0023]** Operational and construction features of the system **115** will be explained in detail successively with respect to different figures. **FIG. 2** illustrates a block diagram of the system **115** for monitoring infrastructure in the cloud network **110**, according to one or more embodiments of the present disclosure.

**[0024]** As per the illustrated embodiment, the system **115** includes one or more processors **205**, a memory **210**, and an input/output interface unit **215**. The one or more processor **205**, hereinafter referred to as the processor **205** may be implemented as one or more microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, state machines, logic circuitries, single board computers, and/or any devices that manipulate signals based on operational instructions. As per the illustrated embodiment, the system **115** includes one or more processors **205**. However, it is to be noted that the system **115** may include multiple processors as per the requirement and without deviating from the scope of the present disclosure. Among other capabilities, the one or more processors **205** are configured to fetch and execute computer-readable instructions stored in the memory **210**. The memory **210** may be configured to store one or more computer-readable instructions or routines in a non-transitory computer-readable storage medium, which may be fetched and executed to create or share data packets over a network service. The memory **210** may include any non-transitory storage device including, for example, volatile memory such as RAM, or non-volatile memory such as EPROM, flash memory, and the like.

**[0025]** In an embodiment, the input/output (I/O) interface unit **215** includes a variety of interfaces, for example, interfaces for data input and output devices, referred to as Input/Output (I/O) devices, storage devices, and the like. The I/O interface unit **215** facilitates communication of the system **115**. In one embodiment, the I/O interface unit **215** provides a communication pathway for one or more components of the system

**115**. Examples of such components include, but are not limited to, a backend database **220** and a distributed cache **225**.

**[0026]** The database **220** is one of, but is not limited to, a centralized database, a cloud-based database, a commercial database, an open-source database, a distributed database, an end-user database, a graphical database, a No-Structured Query Language (NoSQL) database, an object-oriented database, a personal database, an in-memory database, a document-based database, a time series database, a wide column database, a key value database, a search database, a cache database, and so forth. The foregoing examples of the database **220** types are non-limiting and may not be mutually exclusive e.g., a database can be both commercial and cloud-based, or both relational and open-source, etc.

**[0027]** The distributed cache **225** is a pool of Random-Access Memory (RAM) of multiple networked computers into a single in-memory data store for use as a data cache to provide fast access to data. The distributed cache **225** is essential for applications that need to scale across multiple servers or are distributed geographically. The distributed cache **225** ensures that data is available close to where it's needed, even if the original data source is remote or under heavy load.

**[0028]** Further, the processor **205**, in an embodiment, may be implemented as a combination of hardware and programming (for example, programmable instructions) to implement one or more functionalities of the processor **205**. In the examples described herein, such combinations of hardware and programming may be implemented in several different ways. For example, the programming for the processor **205** may be processor-executable instructions stored on a non-transitory machine-readable storage medium and the hardware for the processor **205** may comprise a processing resource (for example, one or more processors), to execute such instructions. In the present examples, the memory **210** may store instructions that, when executed by the processing resource, implement the processor **205**. In such examples, the system **115** may comprise the memory **210** storing the instructions and the processing resource to execute the instructions, or the memory **210** may be separate

but accessible to the system **115** and the processing resource. In other examples, the processor **205** may be implemented by electronic circuitry.

**[0029]** For the system **115** to monitor infrastructure in the cloud network **110**, the processor **205** includes a capturing unit **230** configured to capture a server inventory from the cloud network **110**. The server inventory refers to a comprehensive list including detailed information about all the servers deployed within the cloud network **110**. It serves as a centralized repository that system administrators and other stakeholders can refer to for managing and maintaining servers effectively. Key elements present in the server inventory typically include server identification, hardware details, operating system, network configuration, software and application, and roles and services. The processor **205** further includes an extraction unit **235** configured to extract a plurality of container details from the captured server inventory. The containers in a cloud environment refer to lightweight, portable, and self-contained units of software that package application code and all its dependencies, including libraries and runtime environments, into a single package. These containers are designed to run consistently across different computing environments, whether on-premises or in the cloud. The plurality of container details extracted from the server inventory may include container identification, image details, runtime details, resource allocation, networking, heath and status, environment variables, labels and metadata, security and permission, and dependencies and versions.

**[0030]** The processor **205** further includes an assigning unit **240** configured to assign an Internet Protocol (IP) address to one or more Agent Managers (AMs). The one or more AMs are hosted on at least one dedicated host. The assigned IP address corresponds to a host IP address associated with the at least one dedicated host. The processor **205** further includes a transceiver **245** configured to send the plurality of container details to the one or more AMs. The processor **205** further includes a processing unit **250** configured to process the plurality of container details and server details at the one or more AMs to obtain metric data. The server details typically refer to comprehensive information about physical or virtual servers within a cloud

infrastructure. This server details may include location and identification, hardware details, operating system, network configuration, software applications, security and compliance, and backup and recovery. The metric data is used for monitoring and managing performance of resources of the cloud infrastructure, such as uptime, error rate, compute cost, and requests per minute. The processing unit processes the plurality of container details and the server details by ingesting the metric data at the one or more agent managers from a broker. The ingesting comprises consuming, by the one or more AMs, the metric data from the broker. Consuming the metric data from the broker enables obtaining a file in a pre-defined format. The file comprises validated metrics from the metric data.

**[0031]** The processor **205** further includes an updating unit **255** configured to add or remove host, to or from the at least one dedicated host. The processor **205** further includes a fetching unit **260** configured to fetch, from the broker, the file with the pre-defined format comprising the validated metrics. The processing unit **250** is also configured to process the file received from an infrastructure enrichment unit, Processing the file may include shrinking the metric data contained therein using data normalization. Successively, the metric data is stored in a data lake **120**. The processor **205** further includes a detection unit **265** configured to detect anomalies in the validated metrics stored in the data lake **120**. Anomalies in the metric data refer to deviations or irregularities from expected patterns or norms in various metrics that are monitored within the cloud environment. The anomalies are detected based on comparison of current values of the metric data with historical values of predefined thresholds. For example, a sudden increase or decrease in metrics, such as CPU usage may be identified as an anomaly. Further, a sudden increase in consumption of memory by an application could also be identified as an anomaly.

**[0032]** The processor **205** further includes a triggering unit **270** configured to trigger a feedback unit **302** (illustrated in **FIG. 3**) upon detecting anomalies in the validated metrics stored in the data lake **120**. Triggering of the feedback unit enables reporting and alarm engine to take up a pre-emptive action.

**[0033]** Referring to **FIG. 3** illustrating a block diagram of the environment **100** including the system **115** for monitoring infrastructure in the cloud network **110**, a preferred embodiment of the system **115** is described. The system **115** is alternatively referred as a Cloud-Native Infrastructure Stack (CNIS).

**[0034]** In an embodiment, the system **115** includes a feedback unit **302**, an infrastructure manger **304**, the first host **105-1**, the second host **105-2**, the cloud network **110**, the data lake **120**, a metric ingestion unit **306**, an infrastructure enrichment unit **308**, an infrastructure normalizer unit **310**, and a machine learning/artificial intelligence (AI) unit **312**. The first host **105-1** and the second host **105-2** include at least one agent manager (AM).

**[0035]** The infrastructure manger **304** is connected with the first host **105-1**, the second host **105-2**, and the cloud network **110**. The first host **105-1** and the second host **105-2** also communicate with the metric ingestion unit **306** which is coupled with the infrastructure enrichment unit **308**. The infrastructure enrichment unit **308** is connected with the infrastructure normalizer unit **310**. The machine learning/AI unit **312** is connected with the feedback unit **302**. The machine learning/ AI unit **312**, the feedback engine **302** and the infrastructure normalizer unit **310** are connected with the data lake **120**.

**[0036]** The system **115** incorporates an agentless unique architectural design for efficiently fetching container and host server metrics in a cloud environment. The system **115** includes a dedicated virtual machine (not shown), containerized agent managers (not shown), and an infrastructure manager e.g. dynamic IP assignment unit. Further, the system **115** runs on the dedicated virtual machine and provide a separate and isolated environment for monitoring and managing a cloud infrastructure. The separate and isolated environment allows for better control and resource allocation specific to at least one monitoring tasks.

**[0037]** The system **115** leverages the concept of containerized agent managers. The containerized agent managers are responsible for fetching metrics from containers and the hosts (**105-1** and **105-2**). The containerized agent managers can run multiple agent

managers dynamically so as to enable scalability and resilience in metric collection. The infrastructure manager **304** has the capability to assign internet protocol (IP) addresses on the fly for running the agent managers. The infrastructure manager **304** ensures seamless communication between the agent managers and the monitored containers/hosts (**105-1** and **105-2**) for metric retrieval.

**[0038]** The agent managers deployed by the infrastructure manager **304** are responsible for collecting metrics from the containers and the hosts (**105-1** and **105-2**). The agent managers collect information such as CPU usage, memory utilization, network statistics, and other relevant metrics necessary for monitoring the cloud network **110**. The system **115** provides a centralized management interface to configure and monitor the agent managers. The centralized management interface allows operators to control the number of the agent managers, their deployment, and monitoring parameters in a centralized manner. By using containerization and dynamic IP assignment, the CNIS ensures scalability and flexibility in fetching metrics. The system **115** can adapt to changes in the cloud network **110**. Further, the system **115** can handle varying workloads and easily accommodate new containers or hosts. The system **115** focuses on optimizing the collection of container and host metrics in the cloud environment by utilizing dedicated VMs, containerized agent managers, dynamic IP assignment unit. The design of the system **115** enables efficient and scalable monitoring capabilities tailored to specific requirements of the cloud environments. The system **115** operates on dedicated virtual machines to ensure optimal performance and isolation.

**[0039]** As shown in **FIG. 3**, the agent managers interact with network functions on a southbound interface. In a cloud environment, the network functions refer to capabilities and services provided by an underlying networking infrastructure to facilitate communication, connectivity, security, and management of resources. The network functions may include load balancing, content delivery network, security services, network monitoring and management, and network automation and orchestration. The hosts (**105-1** and **105-2**) integrate over a Transmission Control

Protocol (TCP) interface with an agent manager container. The agent managers obtain all counter metric data from the hosts (**105-1** and **105-2**). All the processes are defined at the agent managers so that the agent managers can process and ingest all metrics data from containers running on the virtual machine. Once the metrics are received from the containers, at least one of the agent managers evaluate the data and send the evaluated data to a broker (not shown).

**[0040]** The infrastructure manger **304** is a central component of a platform (or system (**115**)) which interacts with a graphical user interface (GUI)/dashboard on the southbound and the at least one agent manager on a northbound via a Hypertext Transfer Protocol (HTTP) interface. The infrastructure manger **304** allocates host internet protocols (IPs) to the at least one agent manager which is basically configurable. The infrastructure manger **304** provides a support for a set of APIs through which the host (**115-1** and **115-2**) can be easily provisioned as well. Further, the infrastructure manger **304** can add and remove the hosts (**115-1** and **115-2**) based on the requirement.

**[0041]** The metric ingestion unit **306** consumes data from brokers topics (also referred as a broker). A broker refers to a service or component that facilitates communication, messaging, or event-driven interactions between different components or services within the cloud environment. The metric ingestion unit **306** may create a CSV file for the consumed data, which is being pulled by the infrastructure enrichment unit **308**. The metric ingestion unit **306** validates the metrics obtained from the broker topics and performs the data enrichment. Data enrichment refers to the process of enhancing raw data i.e. the metrics with additional information to make it more valuable, informative, and actionable for analysis, decision-making, and other purposes. This enhancement typically involves adding context, metadata, or supplementary data to the metrics.

**[0042]** Further, the metric ingestion unit **306** fetches/pulls files which are being created by a Digital Asset Management (DAM) metric ingestion. The DAM metric ingestion refers to the process of collecting and integrating various metrics and data

points related to the usage, performance, and analytics of digital assets within a cloud environment. Further, the metric ingestion unit **306** pushes the files to the infrastructure normalizer unit **310** for processing the metric data through the infrastructure enrichment unit **308**. The infrastructure normalizer unit **310** is placed on a path of data flow. The infrastructure normalizer unit **310** intelligently processes the incoming data (metric data) from the metric ingestion unit **306** and the infrastructure enrichment unit **308**. The infrastructure normalizer unit **310** shrinks the metric data through filtering and then stores the data into the data lake **120**. The metric data may be filtered based on several factors including alignment with business goals, used defined thresholds, adjusting granularity of metrics, context based filtering, or any custom filter.

**[0043]** The machine learning/AI unit **312** runs on the data filtered by the infrastructure normalizer unit **310** to find any metric anomalies or trigger forecasting for metrics, as soon as the system **115** finds any new metric. As described above, the anomalies in the metric data refer to deviations or irregularities from expected patterns or norms in various metrics that are monitored within the cloud environment. The anomalies are detected based on comparison of current values of the metric data with historical values of predefined thresholds. For example, a sudden increase or decrease in metrics, such as CPU usage may be identified as an anomaly. Further, a sudden increase in consumption of memory by an application could also be identified as an anomaly.

**[0044]** The machine learning/AI unit **312** sends information associated with the anomalies to a forecasting engine (not shown), a reporting and alarm engine (not shown) and an anomaly detection engine to take a preemptive action for the same. Further, the machine learning/AI unit **312** has a capability to perform network expansion in a closed loop automation.

**[0045]** In one implementation, the pre-emptive action taken to address the anomaly may include scaling of resources. When the anomaly indicates increased resource utilization (e.g., CPU or memory spikes), the resources could be scaled temporarily or

permanently to handle increased demand. This could involve adding more virtual machines, increasing instance sizes, or utilizing auto-scaling capabilities provided by cloud services. Further, load balancing could be performed to address the anomaly. Load balancing would involve implementing load balancing strategies to distribute traffic evenly across multiple instances or servers. This helps prevent overloading of specific resources and ensures consistent performance during spikes in traffic. Further, optimization of resource allocation could be done to address the anomaly. It includes analyzing resource usage patterns and adjusting allocation settings (e.g., CPU shares, memory limits) to optimize resource utilization and minimize waste. Further, data integrity checks could be made to address the anomaly. Data integrity and consistency could be validated when the anomalies are related to data processing or storage. Data integrity checks and audits could be done to ensure that data remains accurate and reliable.

**[0046]** The forecasting engine obtains a request from the machine learning/ AI unit 312 to take the preemptive action using a threshold value set for an operational or performance parameter related to the cloud network. The threshold value is set by a user or the system **115**. The forecasting engine has a capability to perform network expansion based on data trends obtained from the machine learning/AI unit **312**. The network expansion refers to scaling and extending of network infrastructure for accommodating growing needs, increased demand, or new requirements within a cloud-based architecture. The network expansion could be done via horizontal scaling or vertical scaling. Horizontal scaling means adding more resources (e.g., virtual machines, containers) to handle increased workload and traffic. Vertical scaling means increasing the capacity of existing resources (e.g., upgrading instance sizes) to meet performance requirements. In one scenario, the threshold value of CPU utilization may be set as 92%, and when such value is surpassed, the network expansion may be performed. The reporting and alarm engine obtains a request from the machine learning/AI unit **312** to generate an alarm using the threshold value.

**[0047]** **FIG. 4** illustrates a flow chart of a method **400** for monitoring infrastructure in a cloud network, according to one or more embodiments of the present disclosure. For the purpose of description, the method **400** is described with the embodiments as illustrated in **FIGS. 1** and **3** and should nowhere be construed as limiting the scope of the present disclosure. A person of ordinary skill in the art will readily ascertain that the illustrated steps are set out to explain the exemplary embodiments shown, and it should be anticipated that ongoing technological development will change the manner in which particular functions are performed. These examples are presented herein for purposes of illustration, and not limitation. Further, the boundaries of the functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternative boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed. Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the disclosed embodiments.

**[0048]** At step S**402**, an infrastructure manager fetches server inventory from a cloud network. The server inventory refers to a comprehensive list including detailed information about all the servers deployed within the cloud network. It serves as a centralized repository that system administrators and other stakeholders can refer to for managing and maintaining servers effectively. Key elements present in the server inventory typically include server identification, hardware details, operating system, network configuration, software and application, and roles and services.

**[0049]** At step **S404**, the infrastructure manager fetches container details form the servers (e.g., cloud server, edge server, or the like). The containers in a cloud environment refer to lightweight, portable, and self-contained units of software that package application code and all its dependencies, including libraries and runtime environments, into a single package.

**[0050]** At step **S406**, the infrastructure manager distributes containers to at least one agent manager running on dedicated hosts. The agent manager is hosted on a dedicated host.

**[0051]** At step **S408**, the the agent manager fetches metrics/metric data form the containers and the server. The metric data is used for monitoring and managing performance of resources of the cloud infrastructure, such as uptime, error rate, compute cost, and requests per minute.

**[0052]** At step **S410**, the at least one agent manager sends the metric to a metric ingestion unit.

**[0053]** At step **S412**, the metric ingestion unit sends metric data to an infrastructure enrichment unit. The infrastructure enrichment unit sends the metric data to a data lake after normalizing the metric data using an infrastructure normalizer unit.

**[0054]** At step **S414**, an AI/ML unit fetches the metric data to find anomalies by processing the metric data. The AI/ML unit sends a request to a feedback engine for taking a closed loop action, upon identification of an anomaly.

**[0055]** The present invention further discloses a non-transitory computer-readable medium having stored thereon computer-readable instructions. The computer-readable instructions are executed by the processor **205**. The processor **205** is configured to receive capture a server inventory from a cloud. The processor **205** is further configured to extract a plurality of container details from the captured server inventory. The processor **205** is further configured to assign an IP address to one or more agent managers. The processor **205** is further configured to send the plurality of container details to the one or more agent mangers. The processor **205** is further configured to process the plurality of container details and server details at the one or more agent managers to obtain metric data.

**[0056]** A person of ordinary skill in the art will readily ascertain that the illustrated embodiments and steps in description and drawings (**FIGS.1-4**) are set out to explain the exemplary embodiments shown, and it should be anticipated that ongoing

technological development will change the manner in which particular functions are performed. These examples are presented herein for purposes of illustration, and not limitation. Further, the boundaries of the functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternative boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed. Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the disclosed embodiments.

**[0057]** The above described techniques (of monitoring infrastructure in a cloud network) of the present invention provide multiple advantages, including performing multiple actions after the agent manager fetches data from the hosts. Using the above proposed methods, the CNIS incorporates the agentless unique architectural design for efficiently fetching container and host metrics in the cloud network.

**[0058]** The proposed system provides a centralized management interface to configure and monitor the agent managers. The centralized management interface allows operators to control the number of the agent managers, their deployment, and monitoring parameters in a centralized manner. By using containerization and dynamic IP assignment, the CNIS ensures scalability and flexibility in fetching metrics. The system can adapt to changes in the cloud network, and handle varying workloads and easily accommodate new containers or hosts. The system focuses on optimizing the collection of container and host metrics in the cloud network by utilizing dedicated VMs, containerized agent managers, dynamic IP assignment unit. Design of the system enables efficient and scalable monitoring capabilities tailored to specific requirements of the cloud network.

**[0059]** The present invention offers multiple advantages over the prior art and the above listed are a few examples to emphasize on some of the advantageous features. The listed advantages are to be read in a non-limiting manner.

**[0060]** Server: A server may include or comprise, by way of example but not limitation, one or more of a standalone server, a server blade, a server rack, a bank of servers, a server farm, hardware supporting a part of a cloud service or system, a home server, hardware running a virtualized server, one or more processors executing code to function as a server, one or more machines performing server-side functionality as described herein, at least a portion of any of the above, some combination thereof. In an embodiment, the entity may include, but is not limited to, a vendor, a network operator, a company, an organization, a university, a lab facility, a business enterprise, a defence facility, or any other facility that provides content.

**[0061]** System (for example, computing system): A system may include one or more processors coupled with a memory, wherein the memory may store instructions which when executed by the one or more processors may cause the system to perform offloading/onloading of broadcasting or multicasting content in networks. An exemplary representation of the system for such purpose, in accordance with embodiments of the present disclosure. In an embodiment, the system may include one or more processor(s). The one or more processor(s) may be implemented as one or more microprocessors, microcomputers, microcontrollers, edge or fog microcontrollers, digital signal processors, central processing units, logic circuitries, and/or any devices that process data based on operational instructions. Among other capabilities, the one or more processor(s) may be configured to fetch and execute computer-readable instructions stored in a memory of the system. The memory may be configured to store one or more computer-readable instructions or routines in a non-transitory computer readable storage medium, which may be fetched and executed to create or share data packets over a network service. The memory may comprise any non-transitory storage device including, for example, volatile memory such as Random-Access Memory (RAM), or non-volatile memory such as Electrically Erasable Programmable Read-only Memory (EPROM), flash memory, and the like. In an embodiment, the system may include an interface(s). The interface(s) may comprise a variety of interfaces, for example, interfaces for data input and output devices, referred to as input/output (I/O) devices, storage devices, and the like. The interface(s)

may facilitate communication for the system. The interface(s) may also provide a communication pathway for one or more components of the system. Examples of such components include, but are not limited to, processing unit/engine(s) and a database. The processing unit/engine(s) may be implemented as a combination of hardware and programming (for example, programmable instructions) to implement one or more functionalities of the processing engine(s). In examples described herein, such combinations of hardware and programming may be implemented in several different ways. For example, the programming for the processing engine(s) may be processor executable instructions stored on a non-transitory machine-readable storage medium and the hardware for the processing engine(s) may comprise a processing resource (for example, one or more processors), to execute such instructions. In the present examples, the machine-readable storage medium may store instructions that, when executed by the processing resource, implement the processing engine(s). In such examples, the system may include the machine-readable storage medium storing the instructions and the processing resource to execute the instructions, or the machine-readable storage medium may be separate but accessible to the system and the processing resource. In other examples, the processing engine(s) may be implemented by electronic circuitry. In an aspect, the database may comprise data that may be either stored or generated as a result of functionalities implemented by any of the components of the processor or the processing engines.

**[0062]** Computer System: A computer system may include an external storage device, a bus, a main memory, a read-only memory, a mass storage device, communication port(s), and a processor. A person skilled in the art will appreciate that the computer system may include more than one processor and communication ports. The communication port(s) may be any of an RS-232 port for use with a modem-based dialup connection, a 10/100 Ethernet port, a Gigabit or 10 Gigabit port using copper or fiber, a serial port, a parallel port, or other existing or future ports. The communication port(s) may be chosen depending on a network, such a Local Area Network (LAN), Wide Area Network (WAN), or any network to which the computer system connects. The main memory may be random access memory (RAM), or any

other dynamic storage device commonly known in the art. The read-only memory may be any static storage device(s) including, but not limited to, a Programmable Read Only Memory (PROM) chips for storing static information e.g., start-up or basic input/output system (BIOS) instructions for the processor. The mass storage device may be any current or future mass storage solution, which may be used to store information and/or instructions. The bus communicatively couples the processor with the other memory, storage, and communication blocks. The bus can be, e.g. a Peripheral Component Interconnect (PCI) / PCI Extended (PCI-X) bus, Small Computer System Interface (SCSI), universal serial bus (USB), or the like, for connecting expansion cards, drives, and other subsystems as well as other buses, such a front side bus (FSB), which connects the processor to the computer system. Optionally, operator and administrative interfaces, e.g. a display, keyboard, and a cursor control device, may also be coupled to the bus to support direct operator interaction with the computer system. Other operator and administrative interfaces may be provided through network connections connected through the communication port(s). In no way should the aforementioned exemplary computer system limit the scope of the present disclosure.

REFERENCE NUMERALS

**[0063]** Environment – 100;

**[0064]** Host – 105;

**[0065]** Cloud network - 110;

**[0066]** CNIS/System - 115;

**[0067]** Data lake - 120;

**[0068]** One or more processors -205;

**[0069]** Memory – 210;

**[0070]** Input/output interface unit – 215;

**[0071]** Database – 220;

**[0072]** Distributed cache – 225;

**[0073]** Capturing unit – 230;

**[0074]** Extraction unit – 235;

**[0075]** Assigning unit – 240;

**[0076]** Transceiver – 245;

**[0077]** Processing unit – 250;

**[0078]** Updating unit – 255;

**[0079]** Fetching unit – 260;

**[0080]** Detection unit – 265;

**[0081]** Triggering unit – 270;

**[0082]** Feedback unit – 302;

**[0083]** Infrastructure manager – 304;

**[0084]** Metric ingestion unit – 306;

**[0085]** Infrastructure enrichment unit – 308;

**[0086]** Infrastructure normalizer unit – 310; and

**[0087]** Machine Learning/AI unit – 312.

We Claim:

1. A method for monitoring infrastructure in a cloud network (110), the method comprising the steps of:

capturing, by one or more processors (205), a server inventory from the cloud network (110);

extracting, by the one or more processors (205), a plurality of container details from the captured server inventory;

assigning, by the one or more processors (205), an Internet Protocol (IP) address to one or more agent managers;

sending, by the one or more processors (205), the plurality of container details to the one or more agent mangers; and

processing, by the one or more processors (205), the plurality of container details and server details at the one or more agent managers to obtain a metric data.

2. The method as claimed in claim 1, wherein the one or more agent managers are hosted on at least one dedicated host (105).

3. The method as claimed in claim 2, wherein the assigned or allocated IP address corresponds to a host IP address associated with the at least one dedicated host (105).

4. The method as claimed in claim 3, wherein the method comprises adding or removing the host, by the one or more processors (205), to or from the at least one dedicated host (105).

5. The method as claimed in claim 1, wherein processing of the plurality of container details and the server details comprises the step of ingesting, the metric data by the one or more agent managers from a broker.

6. The method as claimed in claim 5, wherein the ingesting comprises consuming, by the one or more processors (205), the metric data from the broker.

7. The method as claimed in claim 6, wherein consuming, by the one or more processors (205), the metric data from the broker enables obtaining a file in a pre-defined format, wherein the file comprises validated metrics from the metric data.

8. The method as claimed in claim 7, comprising fetching, by the one or more processors (205), the file with the pre-defined format comprising the validated metrics.

9. The method as claimed in claim 8, comprising shrinking, by the one or more processors (205), the metric data received from an infrastructure enrichment unit (308), using data normalization, and store the metric data into a data lake (120).

10. The method as claimed in claim 9, comprising detecting, by the one or more processors (205), anomalies in the validated metrics stored in the data lake (120).

11. The method as claimed in claim 10, comprising triggering, by the one or more processors (205), a feedback unit (302) upon detecting anomalies in the validated metrics stored in the data lake (120).

12. The method as claimed in claim 11, wherein triggering of the feedback unit (302), by one or more processors (205), enables reporting and alarm engine to take a pre-emptive action.

13. A system (115) for monitoring infrastructure in a cloud network (110), the system (115) comprising:

a capturing unit (230) configured to capture a server inventory from the cloud network (110);

an extraction unit (235) configured to extract a plurality of container details from the captured server inventory;

an assigning unit (240) configured to assign an Internet Protocol (IP) address to one or more agent managers;

a transceiver (245) configured to send the plurality of container details to the one or more agent managers; and

a processing unit (250) configured to process the plurality of container details and server details at the one or more agent managers to obtain metric data.

14. The system as claimed in claim 13, wherein the one or more agent managers are hosted on at least one dedicated host (105).

15. The system as claimed in claim 13, wherein the assigned or allocated IP address corresponds to a host IP address associated with the at least one dedicated host (105).

16. The system as claimed in claim 14, wherein an updating unit (255) is configured to add or remove host, to or from the at least one dedicated host (105).

17. The system as claimed in claim 13, wherein the processing unit (250) is configured to process the plurality of container details and the server details by ingesting the metric data at the one or more agent managers from a broker.

18. The system as claimed in claim 17, wherein the ingesting comprises consuming, by the one or more agent managers, the metric data from the broker.

19. The system as claimed in claim 18, wherein consuming the metric data from the broker enables obtaining a file in a pre-defined format, wherein the file comprises validated metrics from the metric data.

20. The system as claimed in claim 19, wherein a fetching unit (260) is configured to fetch the file with the pre-defined format comprising the validated metrics.

21. The system as claimed in claim 20, wherein the processing unit (250) is configured to shrink the metric data received from an infrastructure enrichment unit (308), using data normalization, and store the metric data into a data lake (120).

22. The system as claimed in claim 21, wherein a detection unit (265) is configured to detect anomalies in the validated metrics stored in the data lake (120).

23. The system as claimed in claim 22, wherein a triggering unit is configured to trigger a feedback unit (302) upon detecting anomalies in the validated metrics stored in the data lake (120).

24. The system as claimed in claim 23, wherein triggering of the feedback unit (302) enables reporting and alarm engine to take up a pre-emptive action.

25. A non-transitory computer-readable medium having stored thereon computer-readable instructions that, when executed by a processor (205), causes the processor (205) to:

capture a server inventory from a cloud network (110);

extract a plurality of container details from the captured server inventory;

assign an IP address to one or more agent managers;

send the plurality of container details to the one or more agent mangers; and

process the plurality of container details and server details at the one or more agent managers to obtain metric data.

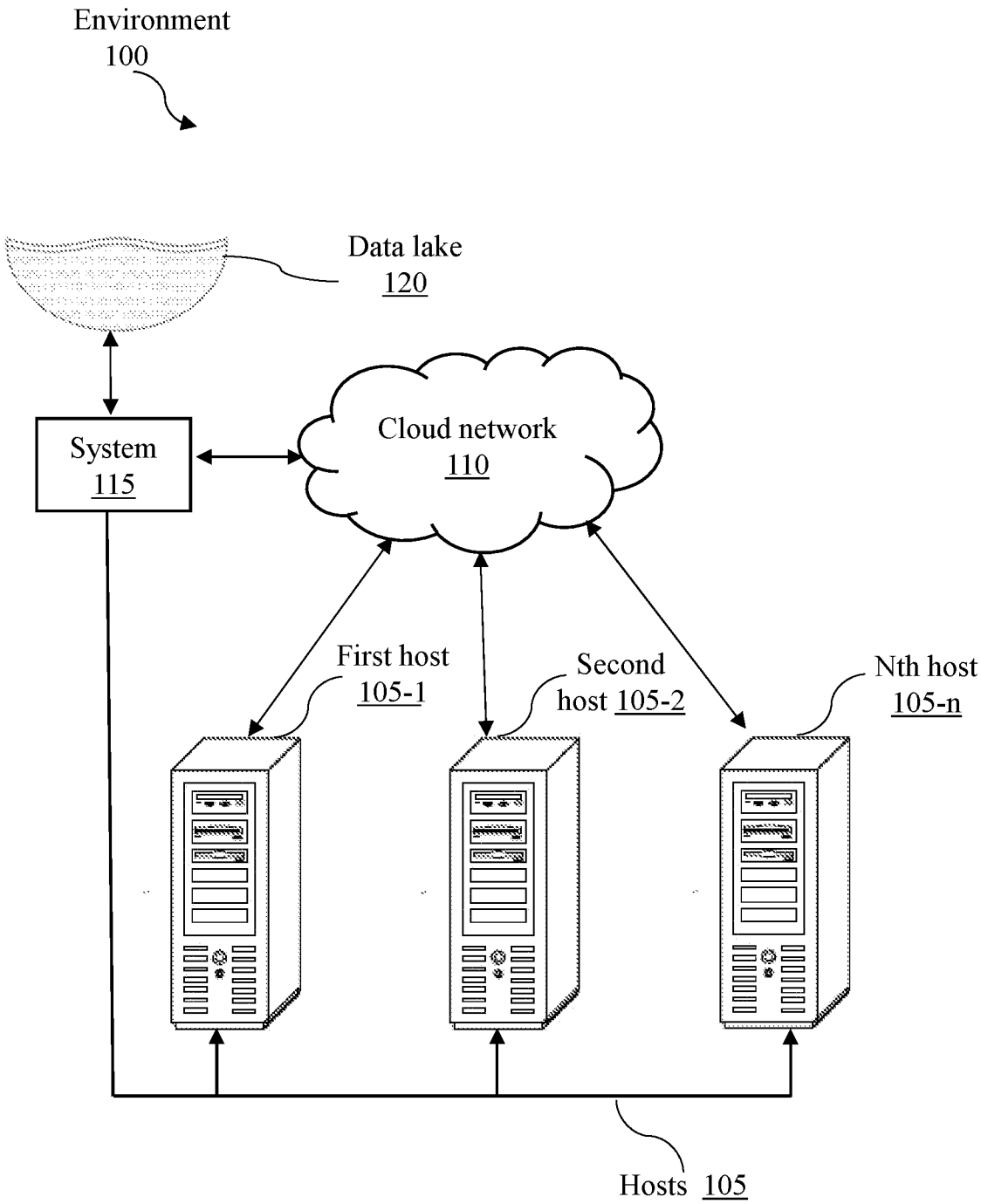
Environment
100
Data lake
120
System
115
Cloud network
110
First host
105-1
Second
host 105-2
Nth host
105-n
Hosts 105

**FIG. 1**

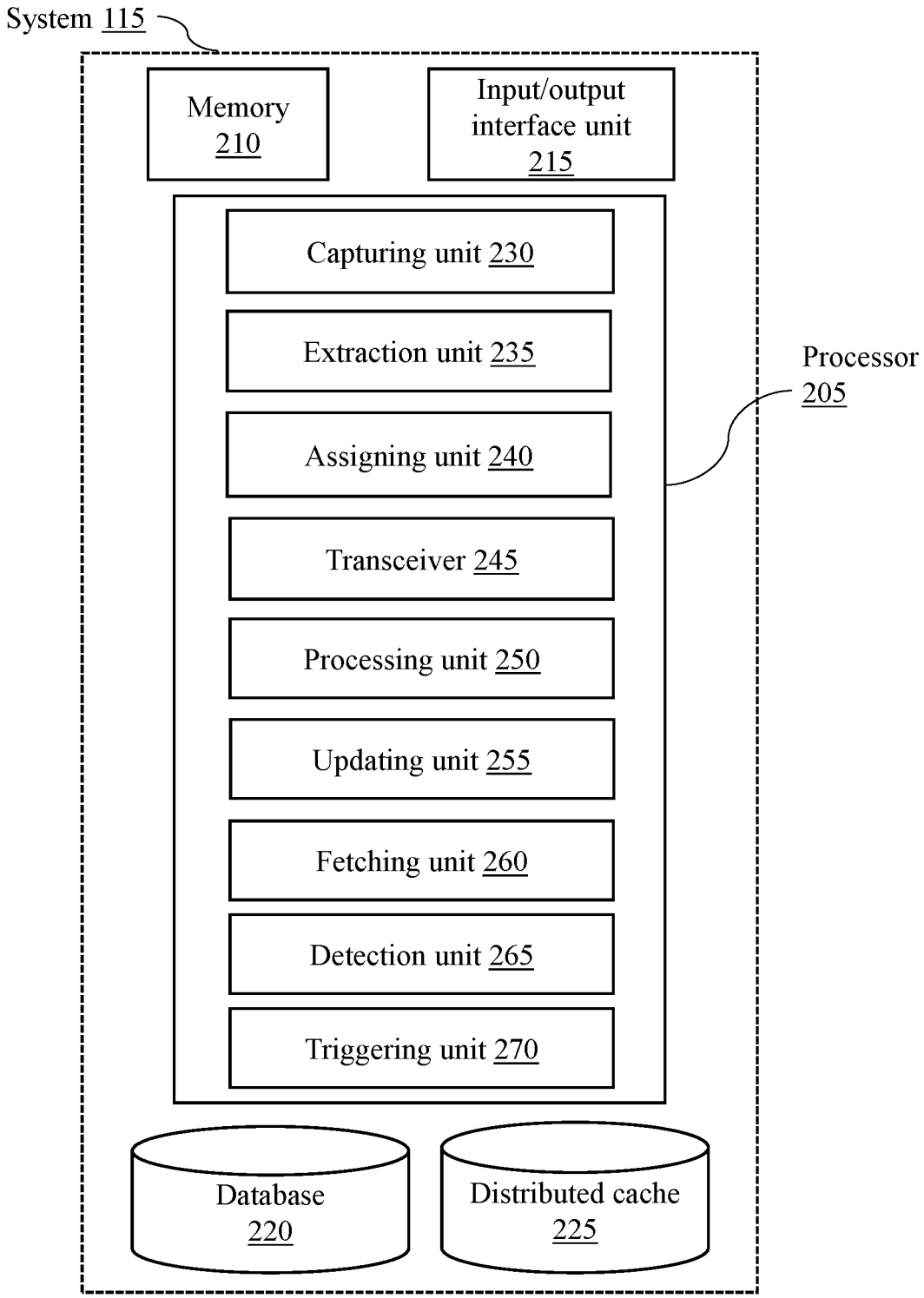
System 115
Memory 210
Input/output interface unit 215
Capturing unit 230
Extraction unit 235
Assigning unit 240
Transceiver 245
Processing unit 250
Updating unit 255
Fetching unit 260
Detection unit 265
Triggering unit 270
Processor 205
Database 220
Distributed cache 225

**FIG. 2**

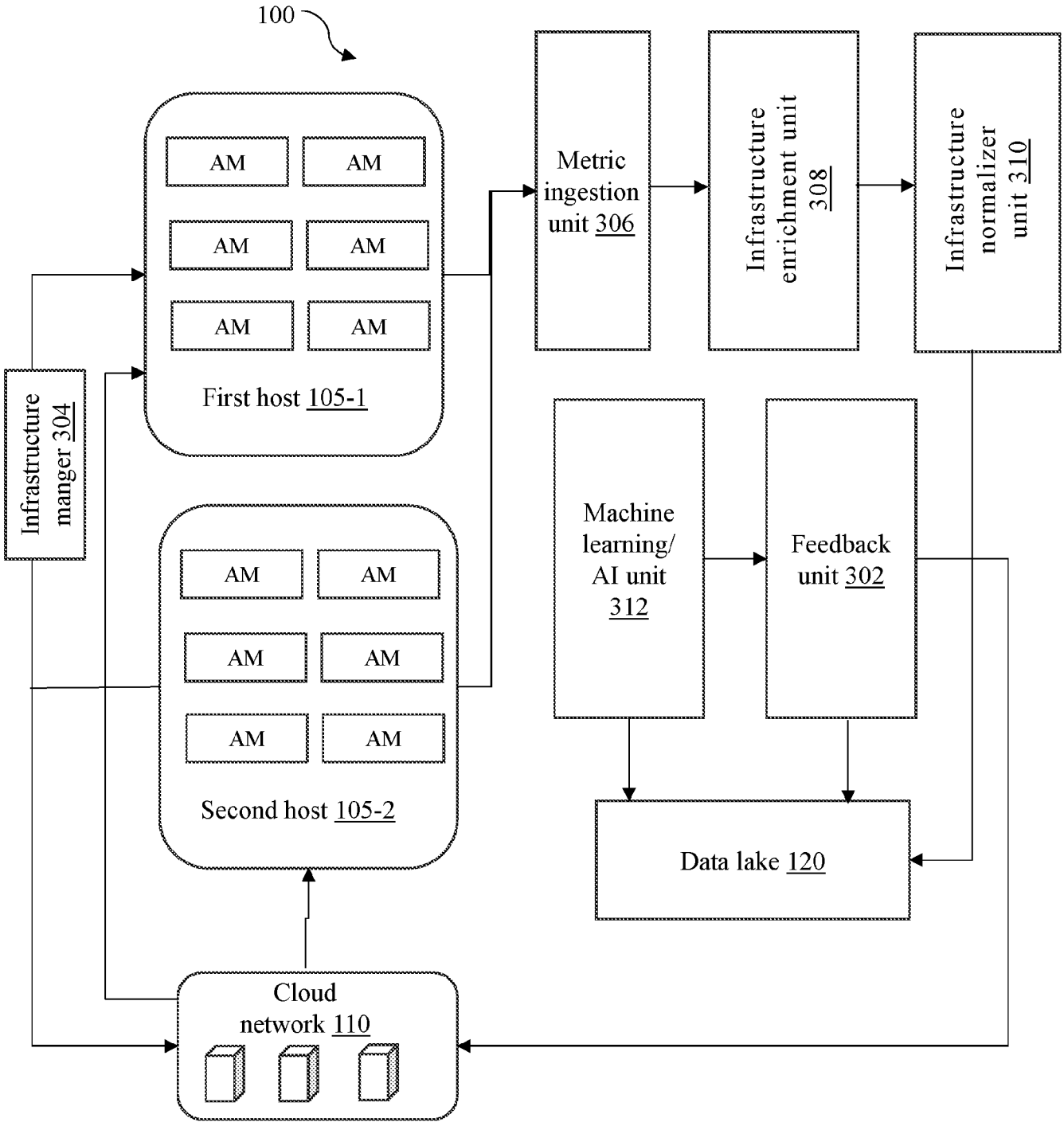
100
AM
AM
AM
AM
AM
AM
First host 105-1
Infrastructure manger 304
Metric ingestion unit 306
Infrastructure enrichment unit 308
Infrastructure normalizer unit 310
Machine learning/ AI unit 312
Feedback unit 302
AM
AM
AM
AM
AM
AM
Second host 105-2
Data lake 120
Cloud network 110

FIG. 3

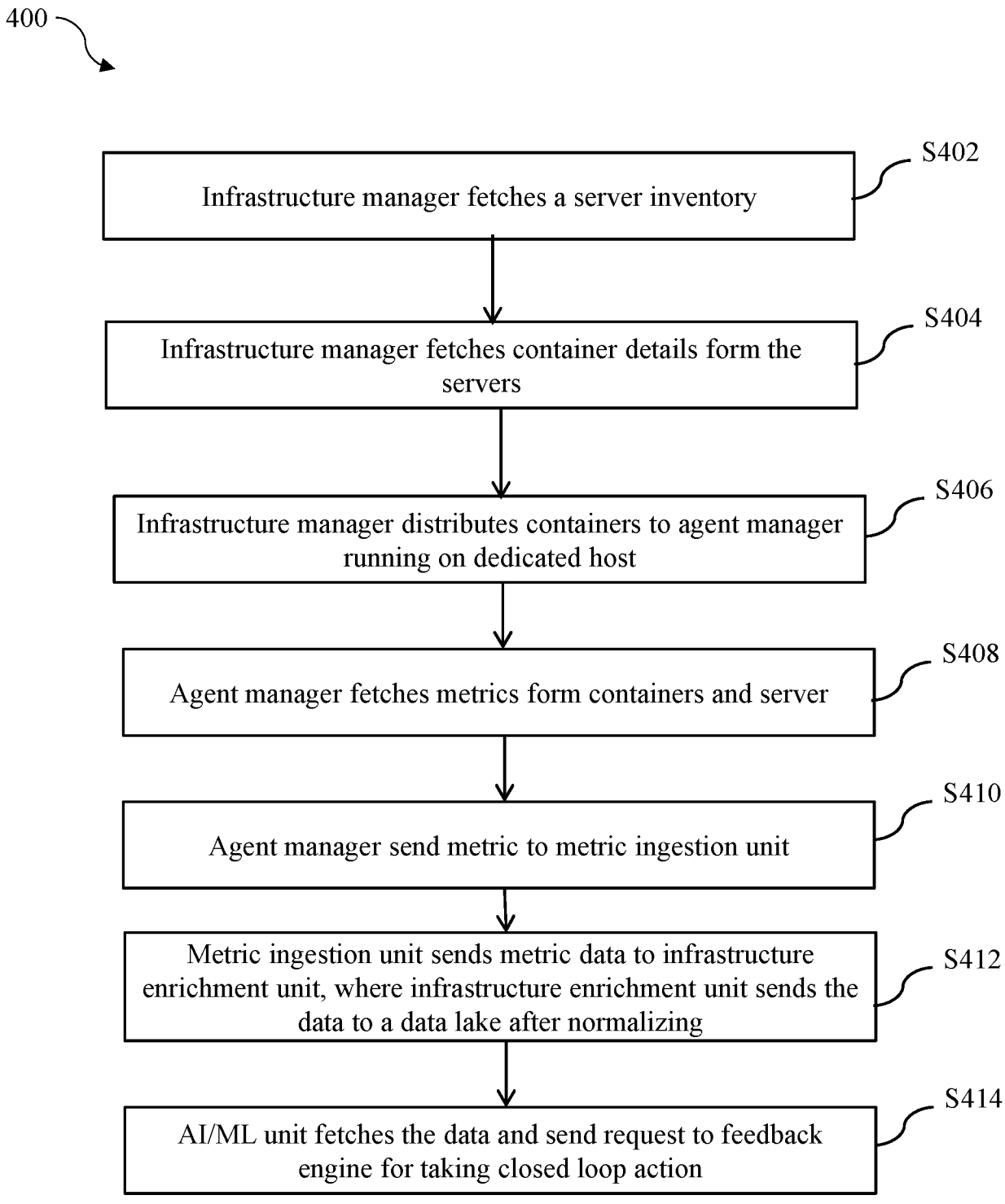
400
Infrastructure manager fetches a server inventory
S402
Infrastructure manager fetches container details form the servers
S404
Infrastructure manager distributes containers to agent manager running on dedicated host
S406
Agent manager fetches metrics form containers and server
S408
Agent manager send metric to metric ingestion unit
S410
Metric ingestion unit sends metric data to infrastructure enrichment unit, where infrastructure enrichment unit sends the data to a data lake after normalizing
S412
AI/ML unit fetches the data and send request to feedback engine for taking closed loop action
S414

**FIG. 4**

# INTERNATIONAL SEARCH REPORT

International application No.
PCT/IN2024/051238

A. CLASSIFICATION OF SUBJECT MATTER
H04L43/08,H04L67/10,H04L41/046 Version=2024.01

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

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

Electronic database consulted during the international search (name of database and, where practicable, search terms used)

Databases: PatSeer, IPO Internal Database
Keywords: Cloud Monitoring, Container, Metric Data, Anomaly, Data Lake

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | Ingest logs, metrics, and uptime data with Elastic Agent; Publication Date: 09 Jan 2022 (09-01-2022) Step 1, Step 2, Step 3; Prerequisites; What's Next? | 1-2,6,12-14,18,24-25 |
| Y | Step 1, Step 2, Step 3; Prerequisites; What's Next?<br>-<br>-<br>- | 3-5,7-11,15-17,19-23 |
| | ---------- | 3-5,7-11,15-17,19-23 |
| Y | WO2022101403A1 (UMNAI LIMITED); Publication Date: 19 May 2022 (19-05-2022) para [0246], [0249]<br>-<br>- | 3-5,7-11,15-17,19-23 |
| | ---------- | |
| Y | MIKELAWS; AppliedInfrastructure/docker-metrics-collector; Publication date: 11 Sept 2016 (11-09-2016) code of Minor update to projcet name in README.md | |

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 10-12-2024 | 10-12-2024 |
| Name and mailing address of the ISA/<br>Indian Patent Office<br>Plot No.32, Sector 14,Dwarka,New Delhi-110075<br>Facsimile No. | Authorized officer<br>Pankaj Kumar Gupt<br>Telephone No. +91-1125300200 |

**INTERNATIONAL SEARCH REPORT**
Information on patent family members

International application No.
PCT/IN2024/051238

| Citation | Pub.Date | Family | Pub.Date |
|---|---|---|---|
| WO 2022101403 A1 | 19-05-2022 | EP 4244783 A1 | 20-09-2023 |
| | | US 11468350 B2 | 11-10-2022 |
| | | US 20230004846 A1 | 05-01-2023 |