US 20240259258A1

(19) **United States**

(12) **Patent Application Publication**

(10) **Pub. No.: US 2024/0259258 A1**

(43) **Pub. Date: Aug. 1, 2024**

REDDY et al.

(54) **GRACEFUL HANDLING OF NORTHBOUND CONNECTIVITY ISSUES IN PERFORMANCE MANAGEMENT AND FAULT MANAGEMENT MICROSERVICES**

(71) Applicant: **Altiostar Networks India Private Limited**, Bangalore, Kamataka (IN)

(72) Inventors: **Virendra REDDY**, Bangalore (IN); **Prasaanth GOWRAVALLI**, Bangalore (IN)

(73) Assignee: **Altiostar Networks India Private Limited**, Bangalore, Kamataka (IN)

(21) Appl. No.: **18/006,464**

(22) PCT Filed: **Nov. 29, 2022**

(86) PCT No.: **PCT/US2022/051244**

§ 371 (c)(1),
(2) Date: **Jan. 23, 2023**

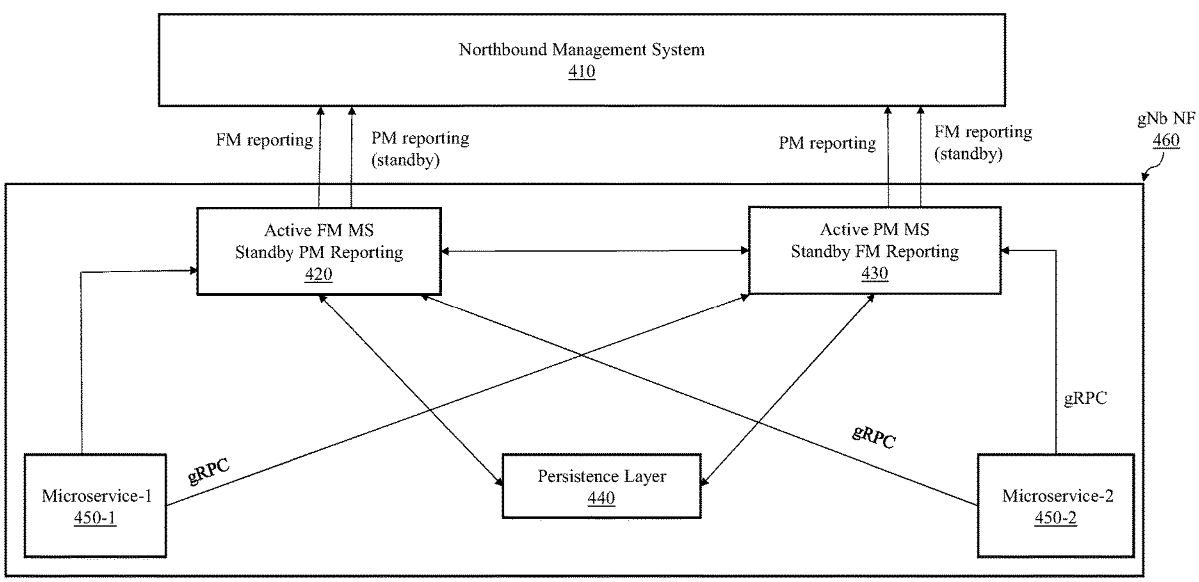
Publication Classification

(51) **Int. Cl.**
H04L 41/069 (2006.01)
H04L 43/0811 (2006.01)

(52) **U.S. Cl.**
CPC ***H04L 41/069*** (2013.01); ***H04L 43/0811*** (2013.01)

(57) **ABSTRACT**

A method and system for reporting data to a northbound management system (NBMS) are provided. The method includes: receiving, by a first microservice (MS) in a Radio Access Network (RAN) Network Function (NF) from at least one other MS of the NF, performance data or event data; determining whether there is a connectivity issue between the first MS and the NBMS; and based on determining that there is the connectivity issue, providing the data to a second MS and reporting, from the second MS to the NBMS, the data, wherein the first MS is one of a Fault Management MS (FM MS) and a Performance Management MS (PM MS), the second MS is another one of the FM MS and the PM MS.



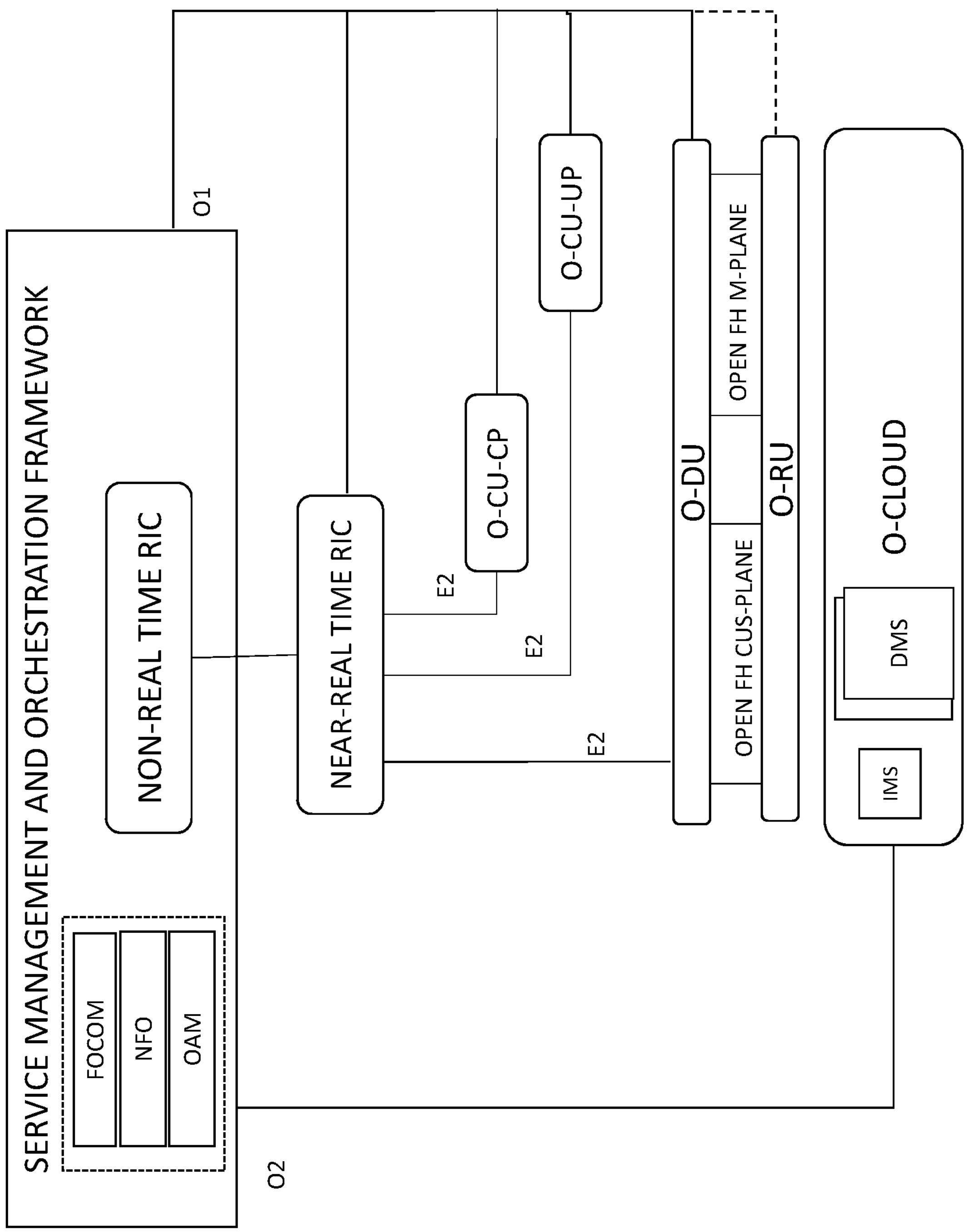
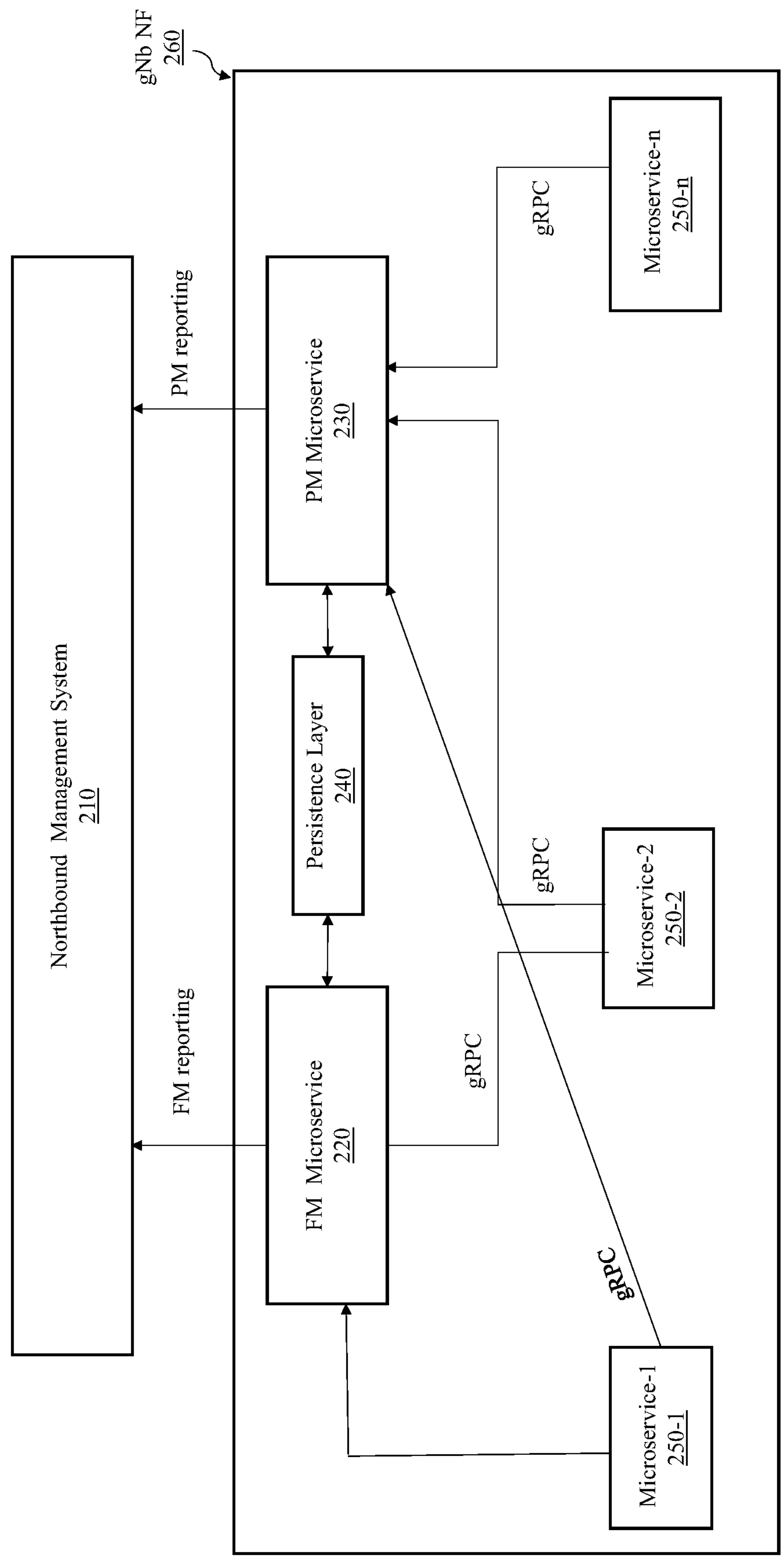

SERVICE MANAGEMENT AND ORCHESTRATION FRAMEWORK
FOCOM
NFO
OAM
NON-REAL TIME RIC
NEAR-REAL TIME RIC
O1
O2
E2
E2
E2
O-CU-CP
O-CU-UP
O-DU
O-RU
OPEN FH M-PLANE
OPEN FH CUS-PLANE
O-CLOUD
DMS
IMS

FIG. 1

FIG. 2


gNb NF
260
Northbound Management System
210
PM reporting
FM reporting
PM Microservice
230
Persistence Layer
240
FM Microservice
220
gRPC
gRPC
gRPC
gRPC
Microservice-n
250-n
Microservice-2
250-2
Microservice-1
250-1

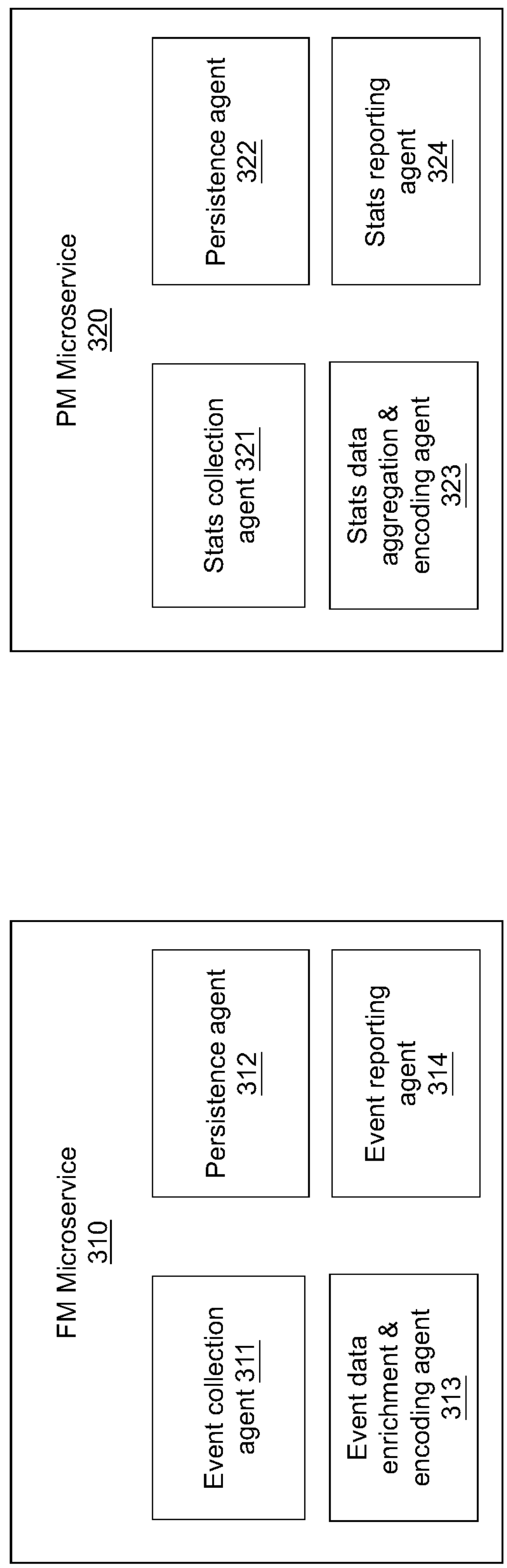


FIG. 3
Related Art

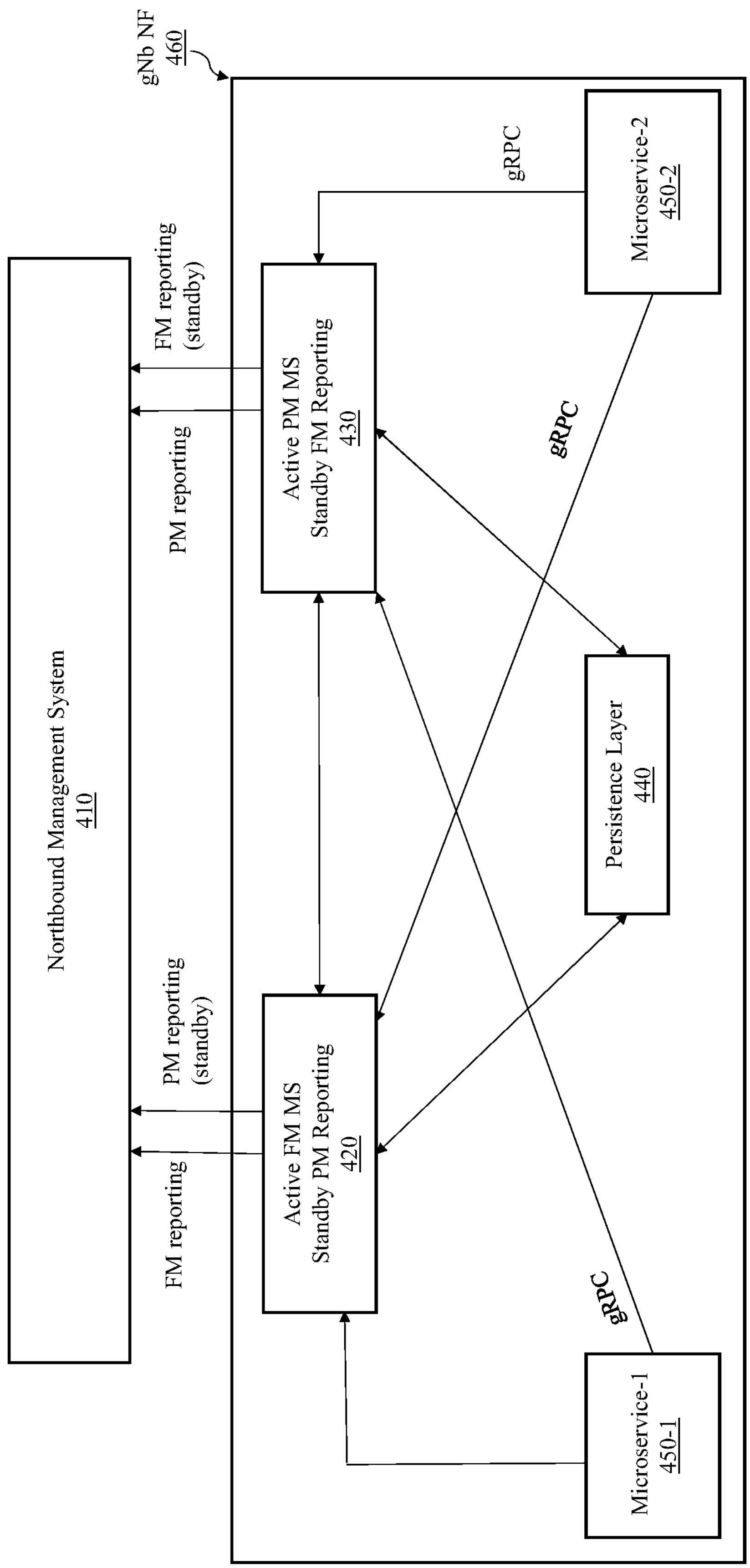

gNb NF
460
Northbound Management System
410
FM reporting
(standby)
PM reporting
PM reporting
(standby)
FM reporting
Active PM MS
Standby FM Reporting
430
Active FM MS
Standby PM Reporting
420
gRPC
Microservice-2
450-2
gRPC
Persistence Layer
440
gRPC
Microservice-1
450-1

FIG. 4

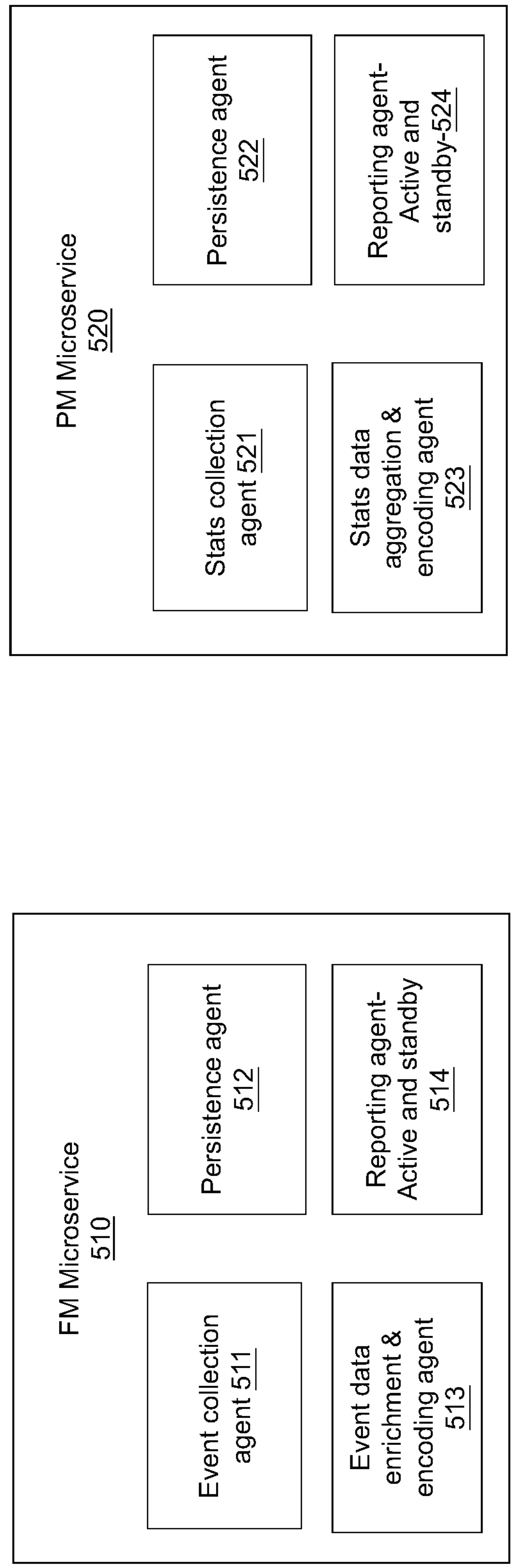

PM Microservice 520
Stats collection agent 521
Persistence agent 522
Stats data aggregation & encoding agent 523
Reporting agent- Active and standby-524
FM Microservice 510
Event collection agent 511
Persistence agent 512
Event data enrichment & encoding agent 513
Reporting agent- Active and standby 514

FIG. 5

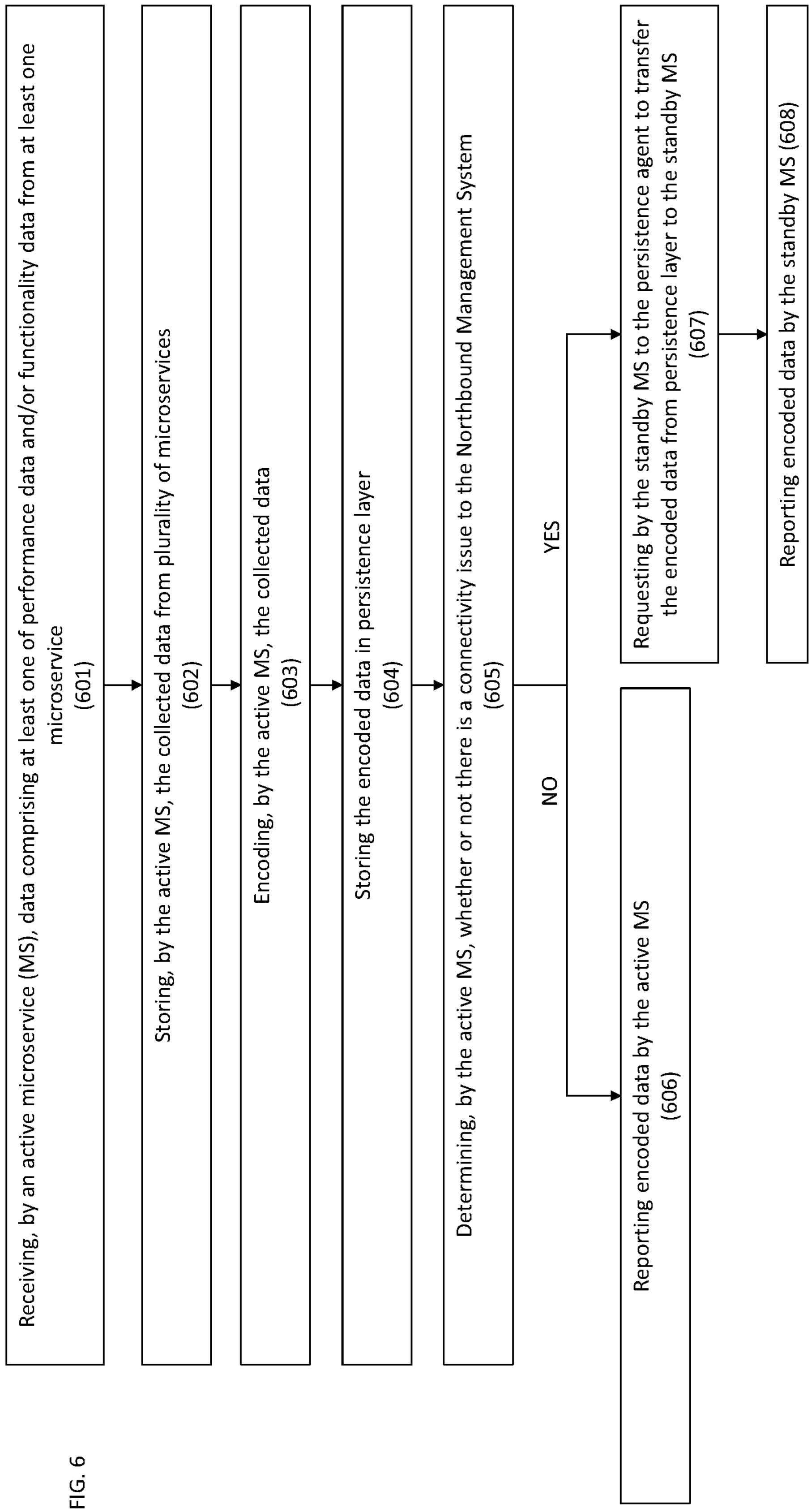
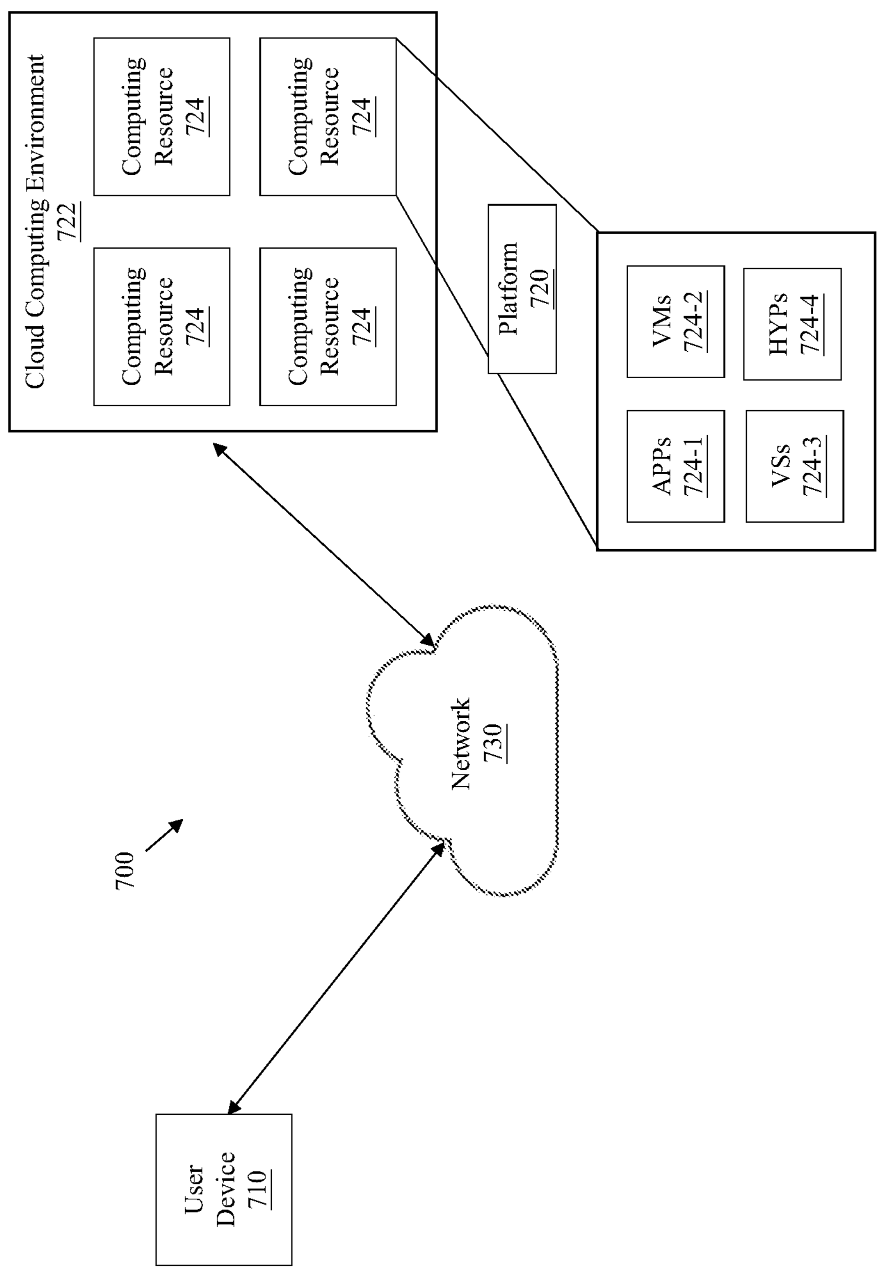

Receiving, by an active microservice (MS), data comprising at least one of performance data and/or functionality data from at least one microservice
(601)
Storing, by the active MS, the collected data from plurality of microservices
(602)
Encoding, by the active MS, the collected data
(603)
Storing the encoded data in persistence layer
(604)
Determining, by the active MS, whether or not there is a connectivity issue to the Northbound Management System
(605)
YES
NO
Requesting by the standby MS to the persistence agent to transfer the encoded data from persistence layer to the standby MS
(607)
Reporting encoded data by the standby MS (608)
Reporting encoded data by the active MS
(606)

FIG. 6

FIG. 7


Cloud Computing Environment 722
Computing Resource 724
Computing Resource 724
Computing Resource 724
Computing Resource 724
Platform 720
APPs 724-1
VMs 724-2
VSs 724-3
HYPs 724-4
Network 730
700
User Device 710

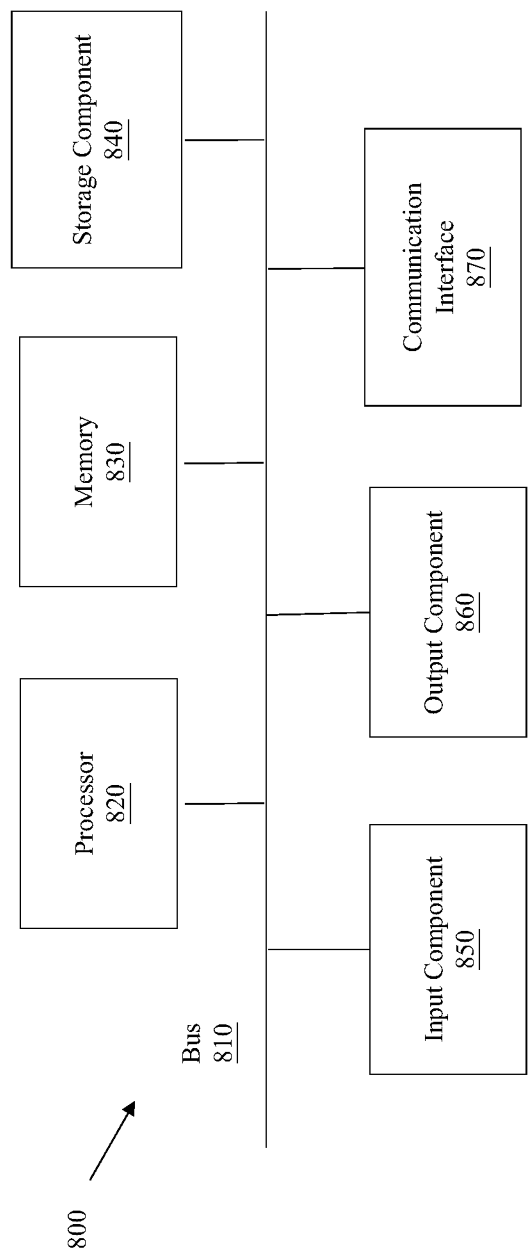

800
Bus
810
Processor
820
Memory
830
Storage Component
840
Input Component
850
Output Component
860
Communication
Interface
870

FIG. 8

GRACEFUL HANDLING OF NORTHBOUND CONNECTIVITY ISSUES IN PERFORMANCE MANAGEMENT AND FAULT MANAGEMENT MICROSERVICES

FIELD

[0001] Systems and methods consistent with example embodiments of the present disclosure relate to handling of northbound connectivity issues in fault and performance management microservices.

BACKGROUND

[0002] A radio access network (RAN) is an important component in a telecommunications system, as it connects end-user devices (or user equipment) to other parts of the network. The RAN includes a combination of various network elements (NEs) that connect the end-user devices to a core network. Traditionally, hardware and/or software of a particular RAN is vendor specific.

[0003] Recently, the evolution of telco technologies enables many telco services to be realized virtually, in the form of software. For instance, RANs such as Open RAN (O-RAN) architectures, disaggregate one network component into multiple functional elements. By way of example, a baseband unit (BBU) or base station (i.e., eNB or gNB) is disaggregated into a number of functional elements including a distributed unit (DU) and a centralized unit (CU), wherein the CU can be further disaggregated into Centralized Unit-Control Plane (CU-CP) and Centralized Unit-User Plane (CU-UP). The disaggregation of network elements enables the telco services and the associated functions to be defined and provided in software-based form or virtual network services, such as Virtualized Network Functions (VNFs), Cloud-native Network Functions (CNFs) or Software Defined Networking (SDN), among others.

[0004] FIG. **1** illustrates a related art O-RAN architecture. Referring to FIG. **1**, RAN functions in the O-RAN architecture are controlled and optimized by a RIC. The RIC is a software-defined component that implements modular applications to facilitate the multivendor operability required in the O-RAN system, as well as to automate and optimize RAN operations. The RIC is divided into two types: a non-real-time RIC (Non-RT RIC) and a near-real-time RIC (Near-RT RIC).

[0005] The Non-RT RIC is the control point of a non-real-time control loop and operates on a timescale greater than 1 second within the Service Management and Orchestration (SMO) framework. Its functionalities are implemented through modular applications called rApps (rApp **1**, . . . , rApp N), and include: providing policy based guidance and enrichment across the A1 interface, which is the interface that enables communication between the Non-RT RIC and the Near-RT RIC; performing data analytics; Artificial Intelligence/Machine Learning (AI/ML) training and inference for RAN optimization; and/or recommending configuration management actions over the O1 interface, which is the interface that connects the SMO to RAN managed elements (e.g., Near-RT RIC, O-RAN centralized Unit (O-CU), O-RAN Distributed Unit (O-DU), etc.).

[0006] The SMO framework manages and orchestrates RAN elements. Specifically, the SMO includes a Federated O-Cloud Orchestration and Management (FOCOM), a Network Function Orchestrator (NFO) that manages Virtual Machines (VM) based VNF and container (i.e., instance) based VNF, and the Operation and Maintenance (OAM) as a part of the SMO that manages and orchestrates what is referred to as the O-Ran Cloud (O-Cloud).

[0007] Further, the SMO may include an operational support system (OSS) and an element management system (EMS), each of which is configurable to perform one or more of: fault management operation, configuration management operation, account management operation, performance management operation, and security management operation (FCAPS operations), on one or more services hosted or deployed in the servers. In some embodiments, the service management system may include a plurality of EMSs, each of the plurality of EMSs may be configured to manage a single service or a group of services associated with a particular vendor/service provider, and the OSS interfaces between the monitoring system, orchestrator, and the plurality of EMSs. Accordingly, the SMO may provide a single control point for managing a plurality of services (associated with multiple vendors/network service providers) via only one monitoring system and one orchestrator system (i.e., one monitoring system and one orchestrator system can be utilized to manage services associated with multiple vendors/service providers).

[0008] In the related art 5G architecture (O-RAN based), functionalities or subsystems in a Network Function (NF) may be implemented as microservices. For example, Radio Unit (RU) manager is a microservice which handles the DU<->RU related transactions. Baseband microservice implements the layer-1 and layer-2 of the 5G stack. FCAPS microservices is a set of microservices which handle aspects of the system like Fault Management (FM), Configuration Management (CM) and Performance Management (PM) of the gNodeB Network Function (gNB NF).

[0009] Fault Management microservice (FM MS) is responsible for collecting various events related to the functionality aspects (e.g., faults, error conditions, etc.) of the 5G O-RAN subsystems (such as those described above) within an NF. The FM MS may collect the corresponding fault events, e.g., cell down, FIC link down, timing locked, etc., from various microservices present in the gNB NF.

[0010] Performance Management microservice (PM MS) is responsible for collecting various statistics and counters related to performance aspects of the 5G O-RAN subsystems (such as those described above) within an NF. It then reports the performance statistics, e.g., throughput, active users, cell uptime, etc., after enriching the data summary to the north bound management system at predefined regular intervals.

[0011] Further, FM MS handles the Fault Management events that are related to functionality aspects, and PM MS handles the Performance Management statistics and counters related to performance aspects. FM MS and PM MS are quite similar microservices with respect to their interaction with the subsystems (collection and persistence aspect) and the northbound (reporting aspect). The FM MS or PM MS, based on predefined conditions, send notification reports to a Northbound Management System (NBMS). The NBMS, based on the notification reports, may create an alarm and consequently, initiates a triggering procedure for post processing or corrective measures that are associated with the alarm.

[0012] FIG. **2** is a block diagram of a communication flow of a Network Function (NF) **260** and a Northbound Management System (NBMS) **210** in the related art. Referring to FIG. **2**, the related art approach is to have a set of FM MS **220** and PM MS **230** in active-active mode to receive, store (in persistence layer **240**), process, and transmit to the NBMS **210** (e.g., EMS and/or OSS) events and statistics received from microservices **250-1**, **250-2**, . . . , **250-*n*** (e.g., microservices implementing various subsystems as described above of the NF) based on predetermined criteria.

[0013] FIG. **3** is a block diagram of components of a Fault Management (FM) microservice **310** and a Performance Management (PM) microservice **320** implemented in an NF. Referring to FIG. **3**, the FM MS **310** may include: event collection agent **311** for receiving event data (e.g., fault-related event data) from subsystems (i.e., microservices) in the NF; persistence agent **312** for storing the received event data (in either or both of raw data form as received or enriched/encoded form as processed by the FM MS **310**); event data enrichment & encoding agent **313** for enriching and encoding the received event data (raw data); and event reporting agent **314** for reporting the enriched and encoded event data to the NBMS **210**. The PM MS **320** may include: stats collection agent **321** for receiving performance statistics data from subsystems (i.e., microservices) in the NF; persistence agent **322** for storing the received performance statistics data (in either or both of raw data form as received or aggregated/encoded form as processed by the PM MS **320**); stats data aggregation & encoding agent **323** for aggregating and encoding the received statistics data (raw data); and stats reporting agent **324** for reporting the aggregated and encoded statistics data to the NBMS **210**.

[0014] In the CNF-based 5G O-RAN implementation, it becomes critical to always have the FM MS and PM MS within a particular NF (e.g., O-CU or O-DU) available in working condition to avoid any impact on the overall functioning of the RAN. To this end, if either of the FM MS or the PM MS is facing northbound connectivity issues, then FM events data or PM statistics data will not be able to be sent northbound. This problem is particularly exacerbated because, in the related art, northbound connectivity issues are typically considered transient and are not deemed as fatal errors so as to trigger a respawn/recreation or switch-over to a standby instance of the microservice(s). This results in loss of critical FM events or PM statistics data and/or a delayed handling of them causing overall user experience issues. Therefore, there is a need to introduce handling of data so that there is no delayed reporting to the NBMS from the FM and PM microservices overall smooth and efficient functioning of RAN in the case of a northbound connectivity issue from either of these microservices **220** and **230**.

SUMMARY

[0015] According to one or more embodiments, apparatuses and methods are provided for graceful handling of northbound connectivity issues by integrating a standby FM MS reporting functionality into a PM MS and integrating a standby PM MS reporting functionality into an FM MS. That is, in accordance with one or more embodiments, the PM MS provides backup reporting functions for the FM MS in case of a northbound connectivity issue for the FM MS, and the FM MS provides backup reporting functions for the PM MS in case of a northbound connectivity issue for the PM MS. As a result, loss or delayed handling of critical FM events or PM statistics data can be avoided, thereby improving overall user experience and network reliability.

[0016] According to at least one embodiment, a method, performed by at least one processor, for reporting data to a northbound management system (NBMS), includes: receiving, by a first microservice (MS) in a Radio Access Network (RAN) Network Function (NF) from at least one other MS of the NF, data comprising performance data or event data related to functionality aspects of the NF; determining whether or not there is a connectivity issue between the first MS and the NBMS; and based on determining that there is the connectivity issue between the first MS and the NBMS, providing the data to a second MS and reporting, from the second MS to the NBMS, the data, wherein the first MS is one of a Fault Management MS (FM MS) and a Performance Management MS (PM MS), wherein the second MS is another one of the FM MS and the PM MS, and wherein the FM MS is configured to receive the event data from the at least one MS, and the PM MS is configured to receive the performance data from the at least one MS.

[0017] According to at least one embodiment, a system for reporting data to a northbound management system (NBMS), includes: at least one memory storing instructions; and at least one processor configured to execute the instructions to: receive, by a first microservice (MS) in a Radio Access Network (RAN) Network Function (NF) from at least one other MS of the NF, data comprising performance data or event data related to functionality aspects of the NF; determine whether or not there is a connectivity issue between the first MS and the NBMS; and based on determining that there is the connectivity issue between the first MS and the NBMS, provide the data to a second MS and report, from the second MS to the NBMS, the data, wherein the first MS is one of a Fault Management MS (FM MS) and a Performance Management MS (PM MS), wherein the second MS is another one of the FM MS and the PM MS, and wherein the FM MS is configured to receive the event data from the at least one MS, and the PM MS is configured to receive the performance data from the at least one MS.

[0018] According to at least one embodiment, a non-transitory computer-readable recording medium has recorded thereon instructions executable by at least one processor to perform a method for reporting data to a northbound management system (NBMS), the method including: receiving, by a first microservice (MS) in a Radio Access Network (RAN) Network Function (NF) from at least one other MS of the NF, data comprising performance data or event data related to functionality aspects of the NF; determining whether or not there is a connectivity issue between the first MS and the NBMS; and based on determining that there is the connectivity issue between the first MS and the NBMS, providing the data to a second MS and reporting, from the second MS to the NBMS, the data, wherein the first MS is one of a Fault Management MS (FM MS) and a Performance Management MS (PM MS), wherein the second MS is another one of the FM MS and the PM MS, and wherein the FM MS is configured to receive the event data from the at least one MS, and the PM MS is configured to receive the performance data from the at least one MS.

[0019] Additional aspects will be set forth in part in the description that follows and, in part, will be apparent from the description, or may be realized by practice of the presented embodiments of the disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] Features, aspects and advantages of certain exemplary embodiments of the disclosure will be described below with reference to the accompanying drawings, in which like reference numerals denote like elements, and wherein:

[0021] FIG. **1** illustrates an O-RAN architecture according to the related art;

[0022] FIG. **2** is a block diagram of a communication flow of a Network Function (NF) and a NorthBound Management System (NBMS) in the related art;

[0023] FIG. **3** is a block diagram of components of a Fault Management (FM) microservice and a Performance Management (PM) microservice implemented in an NF;

[0024] FIG. **4** is a block diagram of a communication flow of a Network Function (NF) and a NorthBound Management System (NBMS) for graceful handling of northbound connectivity issues, according to an embodiment;

[0025] FIG. **5** is a block diagram of components of a FM microservice and a PM microservice, according to an example embodiment;

[0026] FIG. **6** is a flowchart of a method for graceful handling of northbound connectivity issues of an NF, according to an embodiment;

[0027] FIG. **7** is a diagram of an example environment in which systems and/or methods, described herein, may be implemented; and

[0028] FIG. **8** is a diagram of example components of a device according to an embodiment.

DETAILED DESCRIPTION

[0029] The following detailed description of example embodiments refers to the accompanying drawings.

[0030] The foregoing disclosure provides illustration and description, but is not intended to be exhaustive or to limit the implementations to the precise form disclosed. Modifications and variations are possible in light of the above disclosure or may be acquired from practice of the implementations. Further, one or more features or components of one embodiment may be incorporated into or combined with another embodiment (or one or more features of another embodiment). Additionally, in the flowcharts and descriptions of operations provided below, it is understood that one or more operations may be omitted, one or more operations may be added, one or more operations may be performed simultaneously (at least in part), and the order of one or more operations may be switched.

[0031] It will be apparent that systems and/or methods, described herein, may be implemented in different forms of hardware, firmware, or a combination of hardware and software. The actual specialized control hardware or software code used to implement these systems and/or methods is not limiting of the implementations. Thus, the operation and behavior of the systems and/or methods were described herein without reference to specific software code. It is understood that software and hardware may be designed to implement the systems and/or methods based on the description herein.

[0032] Even though particular combinations of features are recited in the claims and/or disclosed in the specification, these combinations are not intended to limit the disclosure of possible implementations. In fact, many of these features may be combined in ways not specifically recited in the claims and/or disclosed in the specification. Although each dependent claim listed below may directly depend on only one claim, the disclosure of possible implementations includes each dependent claim in combination with every other claim in the claim set.

[0033] No element, act, or instruction used herein should be construed as critical or essential unless explicitly described as such. Also, as used herein, the articles "a" and "an" are intended to include one or more items, and may be used interchangeably with "one or more." Where only one item is intended, the term "one" or similar language is used. Also, as used herein, the terms "has," "have," "having," "include," "including," or the like are intended to be open-ended terms. Further, the phrase "based on" is intended to mean "based, at least in part, on" unless explicitly stated otherwise. Furthermore, expressions such as "at least one of [A] and [B]" or "at least one of [A] or [B]" are to be understood as including only A, only B, or both A and B.

[0034] Furthermore, the described features, advantages, and characteristics of the present disclosure may be combined in any suitable manner in one or more embodiments. One skilled in the relevant art will recognize, in light of the description herein, that the present disclosure can be practiced without one or more of the specific features or advantages of a particular embodiment. In other instances, additional features and advantages may be recognized in certain embodiments that may not be present in all embodiments of the present disclosure.

[0035] Example embodiments of the present disclosure provide a method and system for handling of northbound connectivity issues in the FM MS and/or the PM MS. Particularly, when either the FM MS or the PM MS is unable to reach the NBMS due to certain connectivity issues, the active MS that has not lost connectivity may act as a standby MS for reporting to the NBMS. Accordingly, example embodiments of the present disclosure are directed to handling of data so that there is no delayed reporting to the NBMS from the microservices, particularly, FM MS and PM MS, for overall smooth and efficient functioning of the RAN.

[0036] The embodiments of the present disclosure provide a quicker and more efficient approach of handling of reporting of FM events or PM stats data that increases the overall user experience.

[0037] FIG. **4** is a block diagram of a communication flow of a Network Function (NF) **460** (e.g., CU, DU, etc.) and a Northbound Management System (NBMS) **410** for graceful handling of northbound connectivity issues, according to an example embodiment. In particular, FIG. **4** illustrates a system for graceful handling when either the FM MS **420** or the PM MS **430** is having network connectivity issues when reporting data collected through various microservices **450-1**, **450-2**. When an active FM MS **420** or PM MS **430** reports the data to the NBMS **410**, it provides the data to the NBMS **410** (e.g., EMS and/or OSS) through an O1 interface (e.g., management interface). For example, the NBMS **410** may implement a Kafka cluster to collect at least one of FM event data or PM stats data. Kafka is an open-source system that provides a unified, high-throughput, low-latency platform for handling real-time data feeds. Kafka can connect to external systems (for data import/export) via Kafka Connect, and provides the Kafka Streams libraries for stream processing applications. Kafka uses a binary TCP-based protocol that is optimized for efficiency and relies on a "message set" abstraction that naturally groups messages together to reduce the overhead of the network roundtrip. While Kafka may be used in some embodiments, other real time data streams may be used, and these streams may also be open-source. The use of real time data streams, and microservices create a new restructuring open-source platform. The data from FM MS **420** and PM MS **430** may have an Internet Protocol (IP) address (e.g., IPv4, IPv6, etc.) through which it may communicate to the NBMS **410**.

[0038] According to the present disclosure, either FM MS or PM MS may be a standby MS (i.e., include a standby reporting functionality) for the other that is facing network connectivity issues with respect to reporting to NBMS. The standby MS according to an example embodiment will process the other MS's northbound connectivity handling and data reporting in case of Active MS's northbound connectivity issues. Since each microservice can have its own connectivity towards the northbound, therefore, the standby MS would be in working condition when the other one is facing northbound connectivity issues. While in some embodiments, there might be implemented retry and/or reboot mechanisms in order to cure the network issues, relying solely on these mechanisms may lead to further delay in reporting of FM or PM data to NBMS. Therefore, according to an example embodiment, the FM MS **420** or the PM MS **430** may communicate with the other microservice (directly or indirectly via an orchestrator or another component or layer in the NF's architecture, e.g., Kubernetes) that a northbound connectivity issue has been detected and the active MS may handover the data to the other microservice for sending it to the northbound without further delay. In the present embodiment, the active MS may handover enriched/aggregated and encoded data to the other MS (standby MS) via the persistence layer **440**, e.g., by signaling to the other MS that the active MS has a northbound connectivity issue such that the other MS obtains the enriched/aggregated and encoded data from the persistence layer **440**. According to another embodiment, the active MS may directly provide the encoded data to the standby MS.

[0039] As the network issue by the active MS could be faced at any point of time, the raw data, i.e., the event data collected by FM MS or the stats data collected by PM MS, received by the active MS is pushed into or stored in the persistence layer **440** (e.g., the active MS pushes the received raw data into the persistence layer **440** as soon as it receives the raw data). This, for example, allows the active MS to subsequently retrieve the raw data for processing even if the active MS needs to restart or reboot. After this step, the active MS enriches/aggregates and encodes the data, for reporting purposes. To this end, when the active MS has northbound connectivity issues, it hands over (e.g., via the persistence layer **440**) the encoded data to the other (standby) MS to report to the NBMS **410** over its connection to the NBMS **410**.

[0040] Persistence layer **440** can be used to store both raw and encoded data. The data is first stored in persistence layer **440**. The data that is being pushed by microservices is either FM or PM data. In order to make the system resilient to any failure, the raw data may first be pushed to the persistence layer **440** and then pushed back to the respective FM or PM when the MS is active again (e.g., in the case of a reboot or retry mechanism being implemented). The transfer of data from active MS to standby MS for reporting is done when the active MS cannot reach the NBMS **410**. The NBMS **410** may then perform post-processing, reporting, or corrective actions with respect to the data.

[0041] According to an example embodiment as described above, when either the FM MS or the PM MS is unable to reach the NBMS due to certain connectivity issues, the active MS that has not lost connectivity may act as a standby MS for reporting to the NBMS. As a result, the data can be reported to the NBMS without delay, thereby ensuring smooth and efficient functioning of the RAN.

[0042] FIG. **5** is a block diagram of components of a FM microservice **510** and a PM microservice **520**, according to an example embodiment. Referring to FIG. **5**, FM MS **510** with graceful handling of PM MS's **520** northbound connectivity issues may include: event collection agent **511** for receiving event data (e.g., fault-related event data) from subsystems (i.e., microservices) in the NF; persistence agent **512** for storing the received event data (in either or both of raw data form as received or enriched/encoded form as processed by the FM MS **310**); event data enrichment & encoding agent **513** for enriching and encoding the received event data (raw data); and event reporting agent **514** for reporting the enriched and encoded event data to the NBMS, as well as a function of reporting aggregated and encoded PM data (statistics data) in the case that the PM MS **520** has northbound connectivity issues.

[0043] The PM MS **520** may include: stats collection agent **521** for receiving performance statistics data from subsystems (i.e., microservices) in the NF; persistence agent **522** for storing the received performance statistics data (in either or both of raw data form as received or aggregated/encoded form as processed by the PM MS **520**); stats data aggregation & encoding agent **523** for aggregating and encoding the received statistics data (raw data); and stats reporting agent **524** for reporting the aggregated and encoded statistics data to the NBMS, as well as a function of reporting enriched and encoded FM data (event data) in the case that the FM MS **510** has northbound connectivity issues.

[0044] FIG. **6** is a flowchart of a method for graceful handling of northbound connectivity issues of an NF, according to an embodiment. In particular, FIG. **6** illustrates a detailed method for handling of northbound connectivity issues in performance management and fault management microservices. In FIG. **6**, operations **601** through **606** are performed by active MS (one of the FM MS and PM MS) and operations **607** and **608** are performed by standby MS (other one of the FM MS and PM MS that incorporates a standby reporting functionality of the active MS).

[0045] In operation **601**, the active MS receives data comprising at least one performance data and/or event (e.g., fault event, FM data, etc.) data from at least one microservice **450-1**, **450-2**, . . . , **450**-*n*.

[0046] At operation **602**, the active MS stores the received data. For example, the active MS may store the received data in a persistence layer or storage.

[0047] At operation **603**, the active MS encodes the received data. For example, the active MS may retrieve the raw data from storage (e.g., persistence layer) and process (e.g., aggregate, enrich, encode, etc.) the data based on certain prerequisite conditions.

[0048] At operation **604**, the encoded data is pushed or stored in the persistence layer.

[0049] At operation **605**, the active MS determines whether the encoded data may be transmitted to the NBMS, i.e., whether there are any northbound connectivity issues. For example, the active MS may determine that there is a northbound connectivity issue if it cannot establish a connection with the NBMS, if it does not receive an acknowledgement for receipt of previously transmitted data, etc. According to another embodiment, a distinct application, microservice, architecture layer, orchestrator, etc., may determine if there is a northbound connectivity issue with respect to the active MS.

[0050] At operation **606**, if there is no connectivity issue (NO at operation **605**), then the active MS reports the encoded data to the NBMS (e.g., via O1 interface, management interface, etc.).

[0051] At operation **607**, if there is a connectivity issue between the active MS and the NBMS (YES at operation **606**), then the standby MS obtains the encoded data for transmission to the NBMS. For example, the standby MS may be signaled or informed by the active MS of the northbound connectivity issue (or otherwise be requested to transmit the encoded data northbound), and may thereafter request and/or retrieve the encoded data (e.g., from persistence layer).

[0052] At operation **608**, the aforementioned encoded data is reported to NBMS by the standby MS (e.g., via O1 interface, management interface, etc.).

[0053] FIG. **7** is a diagram of an example environment **700** in which systems and/or methods, described herein, may be implemented. As shown in FIG. **7**, environment **700** may include a user device **710**, a platform **720**, and a network **730**. Devices of environment **700** may interconnect via wired connections, wireless connections, or a combination of wired and wireless connections. In embodiments, any of the functions and operations described with reference to FIGS. **4** through **6** above may be performed by any combination of elements illustrated in FIG. **7**.

[0054] User device **710** includes one or more devices capable of receiving, generating, storing, processing, and/or providing information associated with platform **720**. For example, user device **710** may include a computing device (e.g., a desktop computer, a laptop computer, a tablet computer, a handheld computer, a smart speaker, a server, etc.), a mobile phone (e.g., a smart phone, a radiotelephone, etc.), a wearable device (e.g., a pair of smart glasses or a smart watch), or a similar device. In some implementations, user device **710** may receive information from and/or transmit information to platform **720**.

[0055] Platform **720** includes one or more devices capable of receiving, generating, storing, processing, and/or providing information. In some implementations, platform **720** may include a cloud server or a group of cloud servers. In some implementations, platform **720** may be designed to be modular such that certain software components may be swapped in or out depending on a particular need. As such, platform **720** may be easily and/or quickly reconfigured for different uses.

[0056] In some implementations, as shown, platform **720** may be hosted in cloud computing environment **722**. Notably, while implementations described herein describe platform **720** as being hosted in cloud computing environment **722**, in some implementations, platform **720** may not be cloud-based (i.e., may be implemented outside of a cloud computing environment) or may be partially cloud-based.

[0057] Cloud computing environment **722** includes an environment that hosts platform **720**. Cloud computing environment **722** may provide computation, software, data access, storage, etc., services that do not require end-user (e.g., user device **710**) knowledge of a physical location and configuration of system(s) and/or device(s) that hosts platform **720**. As shown, cloud computing environment **722** may include a group of computing resources **724** (referred to collectively as "computing resources **724**" and individually as "computing resource **724**").

[0058] Computing resource **724** includes one or more personal computers, a cluster of computing devices, workstation computers, server devices, or other types of computation and/or communication devices. In some implementations, computing resource **724** may host platform **720**. The cloud resources may include compute instances executing in computing resource **724**, storage devices provided in computing resource **724**, data transfer devices provided by computing resource **724**, etc. In some implementations, computing resource **724** may communicate with other computing resources **724** via wired connections, wireless connections, or a combination of wired and wireless connections.

[0059] As further shown in FIG. **7**, computing resource **724** includes a group of cloud resources, such as one or more applications ("APPs") **724-1**, one or more virtual machines ("VMs") **724-2**, virtualized storage ("VSs") **724-3**, one or more hypervisors ("HYPs") **724-4**, or the like.

[0060] Application **724-1** includes one or more software applications that may be provided to or accessed by user device **710**. Application **724-1** may eliminate a need to install and execute the software applications on user device **710**. For example, application **724-1** may include software associated with platform **720** and/or any other software capable of being provided via cloud computing environment **722**. In some implementations, one application **724-1** may send/receive information to/from one or more other applications **724-1**, via virtual machine **724-2**.

[0061] Virtual machine **724-2** includes a software implementation of a machine (e.g., a computer) that executes programs like a physical machine. Virtual machine **724-2** may be either a system virtual machine or a process virtual machine, depending upon use and degree of correspondence to any real machine by virtual machine **724-2**. A system virtual machine may provide a complete system platform that supports execution of a complete operating system ("OS"). A process virtual machine may execute a single program, and may support a single process. In some implementations, virtual machine **724-2** may execute on behalf of a user (e.g., user device **710**), and may manage infrastructure of cloud computing environment **722**, such as data management, synchronization, or long-duration data transfers.

[0062] Virtualized storage **724-3** includes one or more storage systems and/or one or more devices that use virtualization techniques within the storage systems or devices of computing resource **724**. In some implementations, within the context of a storage system, types of virtualizations may include block virtualization and file virtualization. Block virtualization may refer to abstraction (or separation) of logical storage from physical storage so that the storage system may be accessed without regard to physical storage or heterogeneous structure. The separation may permit administrators of the storage system flexibility in how the administrators manage storage for end users. File virtualization may eliminate dependencies between data accessed at a file level and a location where files are physically stored.

This may enable optimization of storage use, server consolidation, and/or performance of non-disruptive file migrations.

[0063] Hypervisor **724-4** may provide hardware virtualization techniques that allow multiple operating systems (e.g., "guest operating systems") to execute concurrently on a host computer, such as computing resource **724**. Hypervisor **724-4** may present a virtual operating platform to the guest operating systems, and may manage the execution of the guest operating systems. Multiple instances of a variety of operating systems may share virtualized hardware resources.

[0064] Network **730** includes one or more wired and/or wireless networks. For example, network **230** may include a cellular network (e.g., a fifth generation (5G) network, a long-term evolution (LTE) network, a third generation (3G) network, a code division multiple access (CDMA) network, etc.), a public land mobile network (PLMN), a local area network (LAN), a wide area network (WAN), a metropolitan area network (MAN), a telephone network (e.g., the Public Switched Telephone Network (PSTN)), a private network, an ad hoc network, an intranet, the Internet, a fiber optic-based network, or the like, and/or a combination of these or other types of networks.

[0065] The number and arrangement of devices and networks shown in FIG. **7** are provided as an example. In practice, there may be additional devices and/or networks, fewer devices and/or networks, different devices and/or networks, or differently arranged devices and/or networks than those shown in FIG. **7**. Furthermore, two or more devices shown in FIG. **7** may be implemented within a single device, or a single device shown in FIG. **7** may be implemented as multiple, distributed devices. Additionally, or alternatively, a set of devices (e.g., one or more devices) of environment **700** may perform one or more functions described as being performed by another set of devices of environment **700**.

[0066] FIG. **8** is a diagram of example components of a device **800**. Device **800** may correspond to user device **710** and/or platform **720**. As shown in FIG. **8**, device **800** may include a bus **810**, a processor **820**, a memory **830**, a storage component **840**, an input component **850**, an output component **860**, and a communication interface **870**.

[0067] Bus **810** includes a component that permits communication among the components of device **800**. Processor **820** may be implemented in hardware, firmware, or a combination of hardware and software. Processor **820** may be a central processing unit (CPU), a graphics processing unit (GPU), an accelerated processing unit (APU), a microprocessor, a microcontroller, a digital signal processor (DSP), a field-programmable gate array (FPGA), an application-specific integrated circuit (ASIC), or another type of processing component. In some implementations, processor **820** includes one or more processors capable of being programmed to perform a function. Memory **830** includes a random access memory (RAM), a read only memory (ROM), and/or another type of dynamic or static storage device (e.g., a flash memory, a magnetic memory, and/or an optical memory) that stores information and/or instructions for use by processor **820**.

[0068] Storage component **840** stores information and/or software related to the operation and use of device **800**. For example, storage component **840** may include a hard disk (e.g., a magnetic disk, an optical disk, a magneto-optic disk, and/or a solid state disk), a compact disc (CD), a digital versatile disc (DVD), a floppy disk, a cartridge, a magnetic tape, and/or another type of non-transitory computer-readable medium, along with a corresponding drive. Input component **850** includes a component that permits device **800** to receive information, such as via user input (e.g., a touch screen display, a keyboard, a keypad, a mouse, a button, a switch, and/or a microphone). Additionally, or alternatively, input component **850** may include a sensor for sensing information (e.g., a global positioning system (GPS) component, an accelerometer, a gyroscope, and/or an actuator). Output component **860** includes a component that provides output information from device **800** (e.g., a display, a speaker, and/or one or more light-emitting diodes (LEDs)).

[0069] Communication interface **870** includes a transceiver-like component (e.g., a transceiver and/or a separate receiver and transmitter) that enables device **800** to communicate with other devices, such as via a wired connection, a wireless connection, or a combination of wired and wireless connections. Communication interface **870** may permit device **800** to receive information from another device and/or provide information to another device. For example, communication interface **870** may include an Ethernet interface, an optical interface, a coaxial interface, an infrared interface, a radio frequency (RF) interface, a universal serial bus (USB) interface, a Wi-Fi interface, a cellular network interface, or the like.

[0070] Device **800** may perform one or more processes described herein. Device **800** may perform these processes in response to processor **820** executing software instructions stored by a non-transitory computer-readable medium, such as memory **830** and/or storage component **840**. A computer-readable medium is defined herein as a non-transitory memory device. A memory device includes memory space within a single physical storage device or memory space spread across multiple physical storage devices.

[0071] Software instructions may be read into memory **830** and/or storage component **840** from another computer-readable medium or from another device via communication interface **870**. When executed, software instructions stored in memory **830** and/or storage component **840** may cause processor **820** to perform one or more processes described herein.

[0072] Additionally, or alternatively, hardwired circuitry may be used in place of or in combination with software instructions to perform one or more processes described herein. Thus, implementations described herein are not limited to any specific combination of hardware circuitry and software.

[0073] The number and arrangement of components shown in FIG. **8** are provided as an example. In practice, device **800** may include additional components, fewer components, different components, or differently arranged components than those shown in FIG. **8**. Additionally, or alternatively, a set of components (e.g., one or more components) of device **800** may perform one or more functions described as being performed by another set of components of device **800**.

[0074] In embodiments, any one of the operations or processes of FIGS. **4** through **6** may be implemented by or using any one of the elements illustrated in FIGS. **8** and **9**. It is understood that other embodiments are not limited thereto, and may be implemented in a variety of different architectures (e.g., bare metal architecture, any cloud-based architecture or deployment architecture such as Kubernetes, Docker, OpenStack, etc.).

[0075] The foregoing disclosure provides illustration and description, but is not intended to be exhaustive or to limit the implementations to the precise form disclosed. Modifications and variations are possible in light of the above disclosure or may be acquired from practice of the implementations.

[0076] Some embodiments may relate to a system, a method, and/or a computer readable medium at any possible technical detail level of integration. Further, one or more of the above components described above may be implemented as instructions stored on a computer readable medium and executable by at least one processor (and/or may include at least one processor). The computer readable medium may include a computer-readable non-transitory storage medium (or media) having computer readable program instructions thereon for causing a processor to carry out operations.

[0077] The computer readable storage medium can be a tangible device that can retain and store instructions for use by an instruction execution device. The computer readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing. A non-exhaustive list of more specific examples of the computer readable storage medium includes the following: a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), a static random access memory (SRAM), a portable compact disc read-only memory (CD-ROM), a digital versatile disk (DVD), a memory stick, a floppy disk, a mechanically encoded device such as punch-cards or raised structures in a groove having instructions recorded thereon, and any suitable combination of the foregoing. A computer readable storage medium, as used herein, is not to be construed as being transitory signals per se, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide or other transmission media (e.g., light pulses passing through a fiber-optic cable), or electrical signals transmitted through a wire.

[0078] Computer readable program instructions described herein can be downloaded to respective computing/processing devices from a computer readable storage medium or to an external computer or external storage device via a network, for example, the Internet, a local area network, a wide area network and/or a wireless network. The network may comprise copper transmission cables, optical transmission fibers, wireless transmission, routers, firewalls, switches, gateway computers and/or edge servers. A network adapter card or network interface in each computing/processing device receives computer readable program instructions from the network and forwards the computer readable program instructions for storage in a computer readable storage medium within the respective computing/processing device.

[0079] Computer readable program code/instructions for carrying out operations may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, configuration data for integrated circuitry, or either source code or object code written in any combination of one or more programming languages, including an object oriented programming language such as Smalltalk, C++, or the like, and procedural programming languages, such as the "C" programming language or similar programming languages. The computer readable program instructions may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider). In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instructions by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects or operations.

[0080] These computer readable program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks. These computer readable program instructions may also be stored in a computer readable storage medium that can direct a computer, a programmable data processing apparatus, and/or other devices to function in a particular manner, such that the computer readable storage medium having instructions stored therein comprises an article of manufacture including instructions which implement aspects of the function/act specified in the flowchart and/or block diagram block or blocks.

[0081] The computer readable program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other device to cause a series of operational steps to be performed on the computer, other programmable apparatus or other device to produce a computer implemented process, such that the instructions which execute on the computer, other programmable apparatus, or other device implement the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0082] The flowchart and block diagrams in the Figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods, and computer readable media according to various embodiments. In this regard, each block in the flowchart or block diagrams may represent a microservice(s), module, segment, or portion of instructions, which comprises one or more executable instructions for implementing the specified logical function(s). The method, computer system, and computer readable medium may include additional blocks, fewer blocks, different blocks, or differently arranged blocks than those depicted in the Figures. In some alternative implementations, the functions noted in the blocks may occur out of the order noted in the Figures. For example, two blocks shown in succession may, in fact, be executed concurrently or substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts or carry out combinations of special purpose hardware and computer instructions.

[0083] It will be apparent that systems and/or methods, described herein, may be implemented in different forms of hardware, firmware, or a combination of hardware and software. The actual specialized control hardware or software code used to implement these systems and/or methods is not limiting of the implementations. Thus, the operation and behavior of the systems and/or methods were described herein without reference to specific software code-it being understood that software and hardware may be designed to implement the systems and/or methods based on the description herein.

What is claimed is:

1. A method, performed by at least one processor, for reporting data to a northbound management system (NBMS), the method comprising:

receiving, by a first microservice (MS) in a Radio Access Network (RAN) Network Function (NF) from at least one other MS of the NF, data comprising performance data or event data related to functionality aspects of the NF;

determining whether or not there is a connectivity issue between the first MS and the NBMS; and

based on determining that there is the connectivity issue between the first MS and the NBMS, providing the data to a second MS and reporting, from the second MS to the NBMS, the data,

wherein the first MS is one of a Fault Management MS (FM MS) and a Performance Management MS (PM MS),

wherein the second MS is another one of the FM MS and the PM MS, and

wherein the FM MS is configured to receive the event data from the at least one MS, and the PM MS is configured to receive the performance data from the at least one MS.

2. The method of claim **1**, further comprising:

storing, by the first MS, the received data in a persistent storage.

3. The method of claim **1**, further comprising:

processing and/or encoding, by the first MS, the received data,

wherein the providing the data to the second MS comprises, based on determining that there is the connectivity issue between the first MS and the NBMS, providing the processed and/or encoded data to the second MS and reporting, from the second MS to the NBMS, the processed and/or encoded data.

4. The method of claim **3**, wherein the processing comprises at least one of enriching or aggregating the received data.

5. The method of claim **3**, further comprises:

storing, by the first MS, the processed and/or encoded data in a persistent storage,

wherein the providing the data to the second MS comprises, based on determining that there is the connectivity issue between the first MS and the NBMS, providing the processed and/or encoded data from the persistent storage to the second MS.

6. The method of claim **1**, wherein the event data comprises information on at least one fault or error event, and wherein the at least one fault or error event.

7. The method of claim **1**, wherein the performance data comprises at least one performance statistic or counter.

8. A system for reporting data to a northbound management system (NBMS), the system comprising:

at least one memory storing instructions; and

at least one processor configured to execute the instructions to:

receive, by a first microservice (MS) in a Radio Access Network (RAN) Network Function (NF) from at least one other MS of the NF, data comprising performance data or event data related to functionality aspects of the NF;

determine whether or not there is a connectivity issue between the first MS and the NBMS; and

based on determining that there is the connectivity issue between the first MS and the NBMS, provide the data to a second MS and report, from the second MS to the NBMS, the data,

wherein the first MS is one of a Fault Management MS (FM MS) and a Performance Management MS (PM MS),

wherein the second MS is another one of the FM MS and the PM MS, and

wherein the FM MS is configured to receive the event data from the at least one MS, and the PM MS is configured to receive the performance data from the at least one MS.

9. The system of claim **8**, wherein the at least one processor is further configured to execute the instructions to:

store, by the first MS, the received data in a persistent storage.

10. The system of claim **8**, wherein the at least one processor is configured to execute the instructions to:

process and/or encode, by the first MS, the received data; and

based on determining that there is the connectivity issue between the first MS and the NBMS, provide the processed and/or encoded data to the second MS and report, from the second MS to the NBMS, the processed and/or encoded data.

11. The system of claim **10**, wherein the at least one processor is configured to execute the instructions to process the received data by enriching and/or aggregating the received data.

12. The system of claim **10**, wherein the at least one processor is configured to execute the instructions to:

store, by the first MS, the processed and/or encoded data in a persistent storage; and

based on determining that there is the connectivity issue between the first MS and the NBMS, provide the processed and/or encoded data from the persistent storage to the second MS.

13. The system of claim **8**, wherein the event data comprises information on at least one fault or error event, and wherein the at least one fault or error event.

14. The system of claim **8**, wherein the performance data comprises at least one performance statistic or counter.

15. A non-transitory computer-readable recording medium having recorded thereon instructions executable by at least one processor to perform a method for reporting data to a northbound management system (NBMS), the method comprising:

receiving, by a first microservice (MS) in a Radio Access Network (RAN) Network Function (NF) from at least one other MS of the NF, data comprising performance data or event data related to functionality aspects of the NF;

determining whether or not there is a connectivity issue between the first MS and the NBMS; and

based on determining that there is the connectivity issue between the first MS and the NBMS, providing the data to a second MS and reporting, from the second MS to the NBMS, the data,

wherein the first MS is one of a Fault Management MS (FM MS) and a Performance Management MS (PM MS),

wherein the second MS is another one of the FM MS and the PM MS, and

wherein the FM MS is configured to receive the event data from the at least one MS, and the PM MS is configured to receive the performance data from the at least one MS.

16. The non-transitory computer-readable recording medium of claim **15**, wherein the method further comprises:

storing, by the first MS, the received data in a persistent storage.

17. The non-transitory computer-readable recording medium of claim **15**, wherein:

the method further comprises processing and/or encoding, by the first MS, the received data; and

the providing the data to the second MS comprises, based on determining that there is the connectivity issue between the first MS and the NBMS, providing the processed and/or encoded data to the second MS and reporting, from the second MS to the NBMS, the processed and/or encoded data.

18. The non-transitory computer-readable recording medium of claim **17**, wherein the processing comprises at least one of enriching or aggregating the received data.

19. The non-transitory computer-readable recording medium of claim **17**, wherein:

the method further comprises storing, by the first MS, the processed and/or encoded data in a persistent storage; and

the providing the data to the second MS comprises, based on determining that there is the connectivity issue between the first MS and the NBMS, providing the processed and/or encoded data from the persistent storage to the second MS.

20. The non-transitory computer-readable recording medium of claim **15**, wherein:

the event data comprises information on at least one fault or error event; and

the performance data comprises at least one performance statistic or counter.

* * * * *