UNICO I+D Project
6G-INTEGRATION 02

6G-INTEGRATION-02-E14

Monitoring proposal and definition of Key Performance Indicators

Financiado por la Unión Europea
NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia

Document properties

Document number	6G-INTEGRATION-02-E14
Document title	Monitoring Proposal and Definition of Key Performance Indicators
Document responsible	Monica Villalobos, Irene Rubio, Yaima Fiallo (Capgemini Engineering)
Document editor	Yaima Fiallo, Ernesto Correa, Noel Ruiz, Miguel Juaniz, Jon Iriarte (Capgemini Engineering)
Target dissemination level	Private
Status of the document	Draft
Version	1.0
Delivery date	31/12/2024
Other delivery date	31/07/2024 (Partial)

Production properties

Reviewers	Noel Ruiz (Capgemini Engineering)

Disclaimer

This document has been produced in the context of the 6G-INTEGRATION Project. The research leading to these results has received funding from the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union-NextGenerationEU through the UNICO 5G I+D programme.

Contents

List of Figures

List of Tables

List of Acronyms

AI: Artificial Intelligence

CC: Container Capacity

CPUs: Central Processing Units

EO/IR: Electro-Optical/Infrared

GB: Gigabytes

HAPS High Altitude Platforms

IaaS: Infrastructure as a Service

KPIs: Key Performance Indicators

Mbps: Megabits per second

ms: Milliseconds

ML: Machine Learning

NTN: Non-Terrestrial Network

PER: Peripheral Error Rate

PoC: Proof of concept

RAM: Random Access Memory

UAV: Unmanned Aerial Vehicle

UCs: Use Cases

Wh: Watt-hours

1 Introduction

The monitoring of federated nodes represents an innovative strategy in the field of distributed resource management, where various systems collaborate to optimize resource utilization and maximize operational efficiency. This document delves into the data models and definition of Key Performance Indicators (KPIs) necessary to implement effective and eco-friendly resource monitoring.

The primary objective of this analysis is to establish a set of KPIs that enable efficient and sustainable resource management in a federated network. These indicators will cover critical aspects such as geolocation, available energy, resource usage, energy consumption, usage time, and response times. The precise definition of these KPIs is essential to ensure coherence, efficiency, and security in the operation of federated nodes.

Additionally, tools will be proposed to manage monitoring and secure data storage. These tools will facilitate centralized management, real-time monitoring, and resource coordination, ensuring efficient application of federation policies. The document will also include a proposal for persistent data storage, distinguishing between control information and user information.

The Proof of Concept (PoC) developed in previous deliverables will be expanded to incorporate these monitoring tools and defined indicators, aiming to ensure efficient and eco-friendly network management. Tests will be conducted to verify that the network can select the best resources based on these indicators.

Overall, this analysis provides a detailed insight into the KPIs necessary for effective monitoring of federated nodes, exploring their objectives, implementation strategies, essential policies, fundamental requirements, and key tools needed for effective management in federated environments.

2 Monitoring federated nodes

Monitoring federated nodes is essential in distributed networks and systems where multiple independent entities interconnect to share resources and collaborate on data processing. It is a critical component for the efficient, secure, and compliant operation of distributed systems. It is important to highlight that each federated node operates autonomously but collaborates with other nodes in the federated network. Nodes may belong to different organizations and communicate using standard interfaces and protocols to ensure system cohesion.

To perform monitoring, the following criteria should be considered:

- Performance and Availability: Monitoring ensures that federated nodes are functioning correctly, identifying performance issues, and ensuring continuous availability of services.
- Security: Supervising federated nodes' activities is crucial for detecting malicious or unauthorized actions, including monitoring access, intrusion detection, and vulnerability management.
- Propagation Policies: Node owners must comply with specific regulations that mandate monitoring and logging of activities. Defining propagation policies and rules ensures adherence to these regulations.
- Resource Optimization: Through monitoring, bottlenecks and opportunities for resource optimization can be identified, leading to better resource allocation and reduced operational costs.
- Problem Resolution: Effective monitoring facilitates quick identification and resolution of issues. Real-time data enables administrators to diagnose problems and apply solutions before these have a significant impact in the network or users.

Based on these elements, the following objectives for monitoring are defined:

1. Ensure that all nodes operate with the expected efficiency.
2. Ensure that nodes are always accessible and operational.
3. Detect and mitigate security threats in real-time.
4. Verify that data is transmitted and stored correctly.

As shown in the following figure, the structure of the federated network to be used includes a control node, ground station, and Unmanned Aerial Vehicle (UAV) with different elements integrated (NVIDIA, camera, Electro-Optical/Infrared (EO/IR), ARM technology), and requires an integrated and well-structured approach for its monitoring.

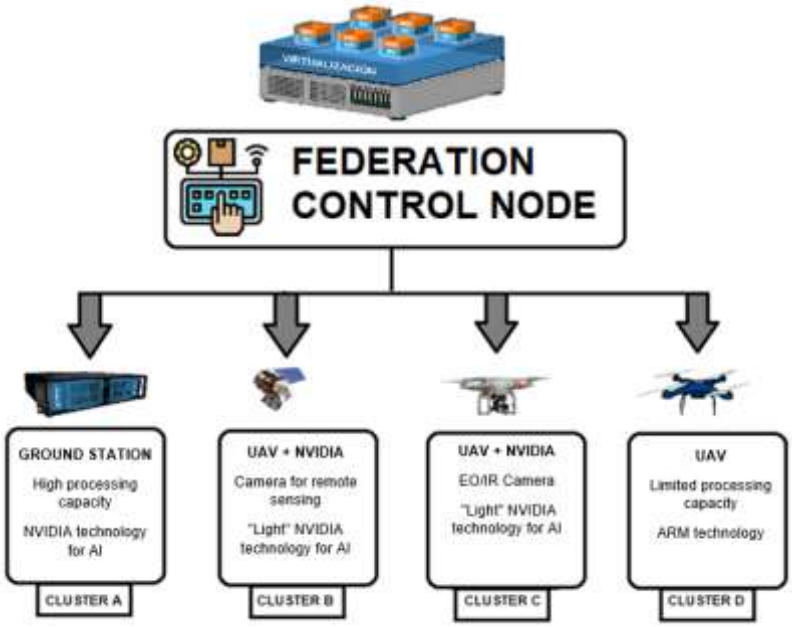

FIGURE 1 FEDERATION SCHEMA

Below is the implementation of monitoring for these federated elements:

Control Node

- Acts as the collection and analysis point for monitoring data.
- Implements a centralized control panel to visualize the status and performance of all federated nodes.
- Sets up an alert and notification system to inform about critical events.

Ground Station

- Implement tools to monitor the hardware (CPU, memory, storage) and software (services, applications) of ground station.
- Uses protocols to transmit state data to the control node.

UAV

- Since UAV may move and operate in remote areas, implementing a remote monitoring system is crucial.
- Equip UAV with telemetry systems to send flight data, battery status, system health, and operational parameters in real-time.
- Use reliable communication links to transmit data to the control node.

The proposed monitoring strategy is based on the following elements (see the next figure):

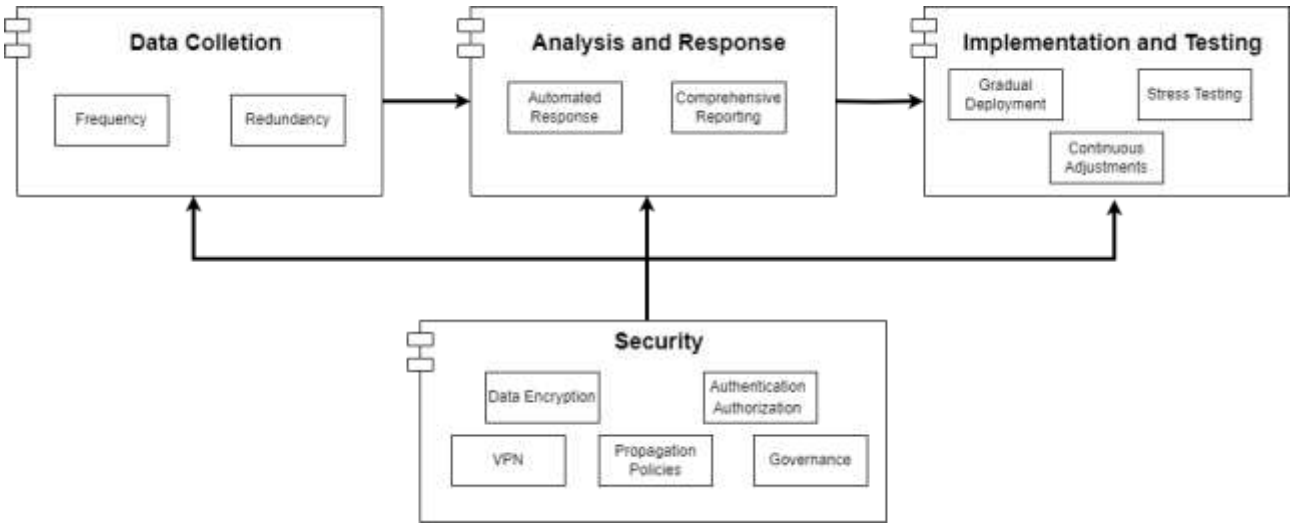

FIGURE 2 MONITORING STRATEGY

Data Collection

- Frequency: Configure the data collection frequency based on the type of node and its criticality. For example, UAV telemetry data every second, state data of ground station every minute
- Redundancy: Ensure redundancy in data collection to avoid loss of critical information

Analysis and Response

- Automated Response: Develop automated response protocols to immediately address detected anomalies or failures. This can include triggering alerts, initiating failover procedures, or executing predefined scripts to mitigate issues.
- Comprehensive Reporting: Generate detailed reports on system performance, anomalies detected, and actions taken. These reports should be accessible to relevant stakeholders for informed decision-making.

Security

- Data Encryption: Ensure that all data transmitted between nodes is encrypted
- Authentication and Authorization: Implement robust authentication and authorization systems to access nodes and monitor data.
- Propagation Policies: In Karmada, provide detailed control over how Kubernetes resources are distributed and managed across multiple clusters. Users can define replication strategies, cluster selection, and configuration customizations to optimize application orchestration in multi-cluster environments.
- Governance: It relates to control and management. In a technology context, "governance" involves establishing rules, policies, and procedures to manage and direct system behavior.
- VPN: Can provide an additional layer of security at the link level. It encrypts all network traffic between network nodes, ensuring secure communication even on a public network. Allows,

as mentioned earlier, deploying a service mesh to orchestrate service and resource sharing between different Kubernetes clusters.

Implementation and Testing

- Gradual Deployment: Implement monitoring in phases, starting with a pilot group of nodes and then scaling to the entire network.
- Stress Testing: Conduct stress tests to ensure that the monitoring system can handle data loads and respond adequately under extreme conditions.
- Continuous Adjustments: Continuously monitor and adjust the monitoring system parameters to optimize performance and reliability.

Effective monitoring of a federated node network requires a combination of advanced tools, robust communication protocols, and real-time analysis strategies. By implementing a well-designed monitoring system, the operability, security, and efficiency of the entire federated network can be ensured, which facilitates proactive management and rapid response to any eventuality.

3 Indicators to facilitate the monitoring of federated nodes

In the evolving landscape of distributed computing, the concept of federated nodes has emerged as a pivotal element in ensuring efficient, scalable, and resilient network operations. Federated nodes, which are independently managed entities collaborating within a unified network framework, require robust monitoring mechanisms to maintain optimal performance and reliability. Effective monitoring of these nodes is essential to pre-emptively identify issues, enhance security, and ensure seamless interoperability among diverse systems.

Federated nodes can span across various geographical locations and operate under different administrative domains, each with its own set of policies, resources, and operational standards. Such diversity necessitates a comprehensive monitoring strategy that can provide real-time insights into the health, performance, and security status of each node. Key indicators for monitoring federated nodes include resource utilization metrics, network latency, throughput, error rates, and security incident reports. These indicators help in identifying bottlenecks, potential failures, and unauthorized access attempts, thereby enabling timely interventions and corrective measures.

Moreover, the integration of advanced technologies such as AI and Machine Learning (ML) can significantly enhance the monitoring capabilities by providing predictive analytics and automated anomaly detection. AI-driven monitoring tools can analyse vast amounts of data from multiple nodes, identify patterns, and predict potential issues before they escalate into major problems. This proactive approach not only improves the overall efficiency of the network but also reduces downtime and enhances user satisfaction.

Furthermore, establishing standardized protocols and frameworks for monitoring federated nodes is crucial for ensuring consistency and interoperability. Organizations can benefit from adopting industry-standard monitoring solutions that offer flexibility and scalability, allowing them to adapt to changing network dynamics and technological advancements. Collaborative efforts among stakeholders to define and implement these standards can lead to more robust and resilient federated networks.

In a Non-Terrestrial Network (NTN), it is essential to monitor various key indicators to ensure optimal performance and efficient management of nodes such as drones, High Altitude Platforms (HAPS), and ground stations. The following table are detailed descriptions of these indicators, their importance, and the units of measurement used.

Indicator	Description	Unit of measurement	Function
Total Random Access Memory (RAM)	Represents the total amount of RAM installed and available on the node	Gigabytes (GB)	Determining the total RAM capacity is crucial for evaluating the node's potential to handle multiple tasks and applications simultaneously. More RAM allows for the execution of more complex applications and the efficient management of larger data volumes
Used RAM	Indicates the amount of RAM currently in use at a given time	GB	Monitoring RAM usage helps detect potential bottlenecks and optimize node performance. High RAM usage may indicate the need for additional RAM or optimization of the running applications
Total Storage Capacity	The total amount of storage space available on the node	GB	Knowing the total storage capacity is essential for planning data loads and ensuring there is enough space for current and future data
Used Storage	The amount of storage space currently in use	GB	Monitoring storage usage helps avoid saturation and plan when additional capacity is needed. It also helps identify usage patterns that can inform optimization decisions
Battery Status	Indicates the amount of remaining energy in the node's battery	Watt-hours (Wh)	This indicator is crucial for mobile nodes such as drones and HAPS, where battery life determines the operational time without recharging. A low battery status can affect service continuity

Total Battery Capacity	The total capacity of the node's battery	Wh	Knowing the total battery capacity helps assess how long the node can operate before needing a recharge. It is vital for planning long-duration missions and operations
Number of Available Central Processing Units (CPUs)	The total number of CPUs available on the node	Integers (number of CPUs)	A higher number of CPUs can improve processing capacity and node performance, allowing more simultaneous tasks and faster data processing
Container Management System Type	The container management system used on the node (e.g., Docker, Kubernetes)	Text (system name)	Identifying available peripherals allows effective management and utilization of these resources, optimizing the node's capabilities for specific applications
Available Peripherals	List of peripherals connected to the node (e.g., cameras, sensors)	Text (names of peripherals)	Identifying available peripherals allows effective management and utilization of these resources, optimizing the node's capabilities for specific applications
Peripheral Status	The current status of peripherals (active, inactive, error)	Text (peripheral status)	Monitoring peripheral status is essential to ensure all node components are functioning correctly. Quickly detecting issues allows corrective actions before they affect operations
Owner	The entity or individual who owns and controls the node	Text (owner name)	Knowing the node's owner is important for administrative management and resolving responsibilities. It facilitates coordination among stakeholders and resource management

Number of Lost Packets	The number of data packets lost during transmission over the network	Integers (number of packets)	Monitoring packet loss is essential for assessing network connection quality and detecting connectivity issues that may affect data transmission
Number of Correct Packets	The number of data packets successfully transmitted over the network	Integers (number of packets)	This indicator helps evaluate the efficiency and reliability of data transmission in the network, ensuring most data is transmitted without errors
Bandwidth Usage	The amount of bandwidth being used on the network	Megabits per second (Mbps)	Knowing bandwidth usage helps manage network capacity, avoid saturation, and ensure optimal performance for applications and services
Latency	The time it takes for a data packet to travel from the source to the destination	Milliseconds (ms)	Latency is a critical indicator of network performance, especially for real-time applications like video or voice transmission. Monitoring latency helps identify and resolve delay issues in the network

TABLE 1 INDICATORS FOR MONITORING FEDERATED NODES

Justification for Monitoring

These indicators were specifically chosen to analyse the status and behaviour of federated nodes within the NTN network. Each indicator provides a scalar value that allows for a quick and precise assessment of various aspects of the node, from CPU and memory performance to battery status and connected peripherals, as well as network performance. This information is vital for ensuring that nodes operate efficiently and reliably, optimizing resource usage, and guaranteeing high-quality service in the network. Continuous monitoring of these parameters allows for rapid problem detection and resolution, performance improvement, and effective planning for the future growth of the network.

4 Definition of KPIs to evaluate the quality of indicators in NTN nodes

To evaluate the quality of the monitored indicators in nodes of a NTN, it is essential to define KPIs that enable a quantitative interpretation of the collected data. Below are the KPIs, formulas, and metrics corresponding to each of the previously defined indicators.

KPIs	Formula	Interpretation
RAM Utilization	$\text{RAM Utilization (\%)} = \frac{\text{Used RAM(GB)}}{\text{Total RAM(GB)}} * 100$	This KPI measures the percentage of RAM used compared to the total available memory. A high value may indicate the need to optimize applications or increase memory capacity
Storage Utilization	$\text{Storsge Utilization (\%)} = \frac{\text{Used Storage(GB)}}{\text{Total Storage(GB)}} * 100$	This KPI measures the percentage of storage used compared to the total available storage. It helps prevent storage saturation and plan for future capacity needs
Battery Status	$\text{Battery Status(\%)} = \frac{\text{Remaining Energy}}{\text{Total Battery Capacity}} * 100$	This KPI shows the percentage of remaining energy in the battery. A low value indicates that the node will soon need recharging, which is crucial for the operation of mobile nodes
CPU Utilization	$\text{CPU Utilization (\%)} = \frac{\text{CPU Time Used}}{\text{Total CPU Time Available}} * 100$	This KPI measures the percentage of CPU time used relative to the total available CPU time. A high percentage may indicate that the node is overloaded and might need

		additional CPUs or a more balanced load distribution
Network Efficiency	$\text{Packet Loss (\%)} = \frac{\text{Number Lost Packets}}{\text{Total Number Packets Sent}} * 100$ $\text{Packet Success Rate (\%)} = \frac{\text{Number Correct Packets}}{\text{Total Number Packets Sent}} * 100$	• Packet Loss (%): A high percentage indicates problems with connection quality, which could suggest network congestion, interference, or other issues affecting data transmission. • Packet Success Rate (%): A high percentage indicates an efficient and reliable network, where most packets are successfully transmitted without errors
Bandwidth Utilization	$\text{Bandwidth Utilization (\%)} = \frac{\text{Used Bandwidth}}{\text{Total Bandwidth}} * 100$	This KPIs measures the percentage of bandwidth used. A high value may indicate the need to increase network capacity or optimize traffic
Latency	$\text{Average Latency (ms)} = \frac{\sum \text{Packet Travel Times(ms)}}{\text{Number of Packet}}$	Average latency measures the travel time of packets. High values can indicate network performance issues, affecting real-time applications
Peripheral Availability (PA)	$\text{PA (\%)} = \frac{\text{Number Active Peripherals}}{\text{Total Number Peripherals}} * 100$	This KPI measures the percentage of peripherals that are active and functioning correctly. A low value may indicate failures in peripherals that need attention

Peripheral Error Rate (PER)	$\text{PER}(\%) = \frac{\text{Number Peripherals with Errors}}{\text{Total Number Peripherals}} * 100$	This KPI measures the percentage of peripherals that are in an error state. A high percentage indicates serious issues that need resolution
Container Capacity (CC)	$\text{CC}\ (\%) = \frac{\text{Number Active Containers}}{\text{Total Number Available Containers}} * 100$	This KPI measures the percentage of active containers compared to the total available. It helps evaluate deployment capacity and resource utilization

TABLE 2 KPIS TO EVALUATE THE QUALITY OF INDICATORS IN NTN NODES

Justification for Monitoring

These KPIs and metrics provide a solid foundation for evaluating performance and the quality of monitored indicators in NTN nodes. Each formula allows for quantitative interpretation, facilitating problem detection, resource optimization, and informed decision-making. Implementing these KPIs ensures that nodes operate efficiently, reliably, and sustainably, optimizing resource use and ensuring high service quality across the network. Constant monitoring and analysis of these parameters enable performance improvement and effective planning for the future growth of the network.

5 Tools to facilitate monitoring management

Effective monitoring of critical variables such as CPU usage, memory utilization, system state, and disk usage is essential for ensuring smooth operations, preventing downtime, and optimizing resource allocation in IT infrastructure. In today's fast-paced and technology-driven environment, maintaining the health and performance of IT systems is paramount. Without proper monitoring, systems can experience performance degradation, unexpected failures, and inefficient use of resources, leading to significant operational challenges and financial losses.

Monitoring tools play a crucial role in providing visibility into the performance and health of IT infrastructure. These tools collect and analyze data in real-time, allowing for immediate detection of issues and facilitating proactive maintenance. By generating alerts and detailed analytics, they enable IT teams to quickly address potential problems before they escalate, ensuring continuous availability and optimal performance of critical systems.

Numerous tools have been developed to excel in monitoring tasks, each offering unique features and capabilities tailored to different environments and requirements. These tools not only provide real-time insights but also help in long-term capacity planning and trend analysis. With advanced visualization options, customizable dashboards, and integration capabilities, modern monitoring solutions empower organizations to make informed decisions, improve resource management, and enhance overall operational efficiency.

In this context, the selection of the right monitoring tools is crucial. The following table presents some tools showcasing their strengths and applications for monitoring these aspects across various environments.

Tool	Description	Key features
Prometheus [1] [2]	Open-source monitoring and alerting toolkit designed specifically for reliability and scalability. It is known for its powerful data model and robust querying language, PromQL, which enables users to extract meaningful insights from the collected metrics. Excels in environments that require high flexibility and detailed custom metrics, making it ideal for modern cloud-native architectures.	• Time series data model, stores metrics with labels, can handle thousands of metrics efficiently. • Powerful query language (PromQL) for extracting and analysing metrics. • Node Exporter for hardware and OS metrics like CPU, memory, disk, and state. • Alert Manager to handle alerts and notifications based on user-defined rules, allowing for timely notifications and remediation.

		• Grafana integration for creating detailed dashboards and visualizations.
Nagios [3]	Powerful open-source tool well-established in the field of infrastructure monitoring. It is widely used for its versatility and comprehensive plugin ecosystem, which supports monitoring a wide range of system metrics and services. Ideal for organizations that require a robust, customizable solution capable of monitoring diverse and complex IT environments.	• Extensive plugin library for monitoring different services and metrics (e.g., CPU load, memory usage, disk usage, network traffic) • Customizable alerts via email, SMS, and other methods. • Performance data graphing through plugins like PNP4Nagios. • Flexible configuration for monitoring a wide range of applications and infrastructure components. • Web-based interface for easy management and reporting. • Strong community and commercial support options provide extensive resources for troubleshooting and optimization.
Zabbix [4]	Open-source monitoring solution known for its high performance and scalability. It is designed to monitor and track the status of various network services, servers, and other network hardware. Particularly suitable for large-scale enterprise environments that demand high availability and detailed performance insights.	• Agent-based and agentless monitoring for collecting detailed metrics (CPU, memory, disk usage). • Auto-discovery for detecting and monitoring devices and services. • Customizable alerts with advanced notification capabilities. • Built-in graphing and dashboards for visualizing performance data • Scalability, it is suitable for large environments with distributed monitoring capabilities with a central management console. • Advanced visualization options with customizable dashboards and detailed reporting.

Datadog [5]	Cloud-based monitoring and analytics platform that provides comprehensive visibility into infrastructure and applications. It is known for its ease of use and integration with numerous cloud services and technologies. Ideal for organizations leveraging cloud services and seeking a seamless, integrated monitoring solution that offers deep insights and predictive capabilities.	• Unified Monitoring combines infrastructure monitoring, application performance monitoring (APM), and log management. • Provides real-time monitoring and analytics with a rich set of visualization tools. • Uses machine learning to detect anomalies and provide predictive analytics. • Scalability, easily scales to monitor thousands of hosts and services across multiple cloud environments
SolarWinds Server & Application Monitor (SAM) [6]	Provides comprehensive monitoring for server and application performance, including CPU, memory, state, and disk usage. Ideal for comprehensive monitoring and management of server and application performance in complex IT environments. Particularly beneficial for organizations that need to ensure high availability and performance of their critical applications and servers, improve operational efficiency, and maintain a reliable IT infrastructure.	• Out-of-the-box templates for monitoring common applications and services. • Customizable alerts and notifications to keep you informed of critical issues. • Performance analysis dashboards for visualizing metrics and trends. • Application dependency mapping to understand the relationships between services. • Historical data analysis to identify performance trends and capacity planning.

TABLE 3 SUMMARY OF MONITORING TOOLS

5.1. Tools to use in PoC

After analyzing different monitoring frameworks we have come to the conclusion that, although they are very powerful frameworks, they are also complex to implement and depend a lot on the operating system and the environment configuration. Taking into account that our environment is limited, and the nodes include drones with low payload, it would be convenient to deploy a simple monitor, easy to implement automatically and independent of the platform. Given that a federated node requires containerization using Kubernetes or Docker, the best approach is to also containerize our monitoring system. This will enable automatic deployment.

A simple version of a monitor, which will be seen later, has been developed in Python with the use of proprietary libraries that do not overload the image and can be deployed in all nodes that are part of the federation, including drones with their small payload.

The monitoring architecture is based on a central control node that hosts an MQTT server (broker) in charge of centralizing the communication. Each federated cluster deploys a monitor that connects to the broker and sends periodic metrics, such as performance and status data, at regular intervals. These metrics are collected by the central node, which processes them for various purposes: displaying them on a dashboard to provide a consolidated view of the system or using them for analysis and decision making, such as identifying the most suitable node for future deployments. In this way, the system ensures efficient and scalable monitoring, while maintaining the autonomy of the federated clusters.

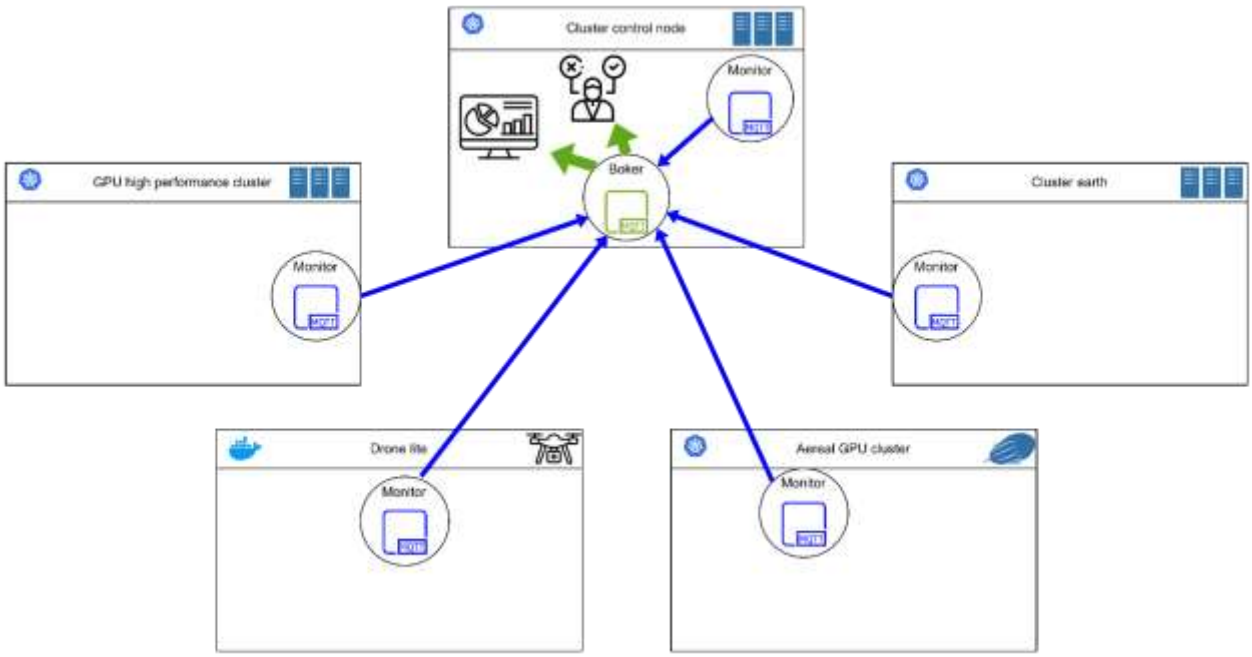

FIGURE 3 MONITORING SCHEMA

6 PoC to include monitoring tools

As federated systems continue to evolve, efficient resource management becomes a critical element for the success of these infrastructures. The PoC developed in deliverable 6G-INTEGRATION-02-E13 implemented an application layer security strategy using a VPN, an identity and access management system implementing Keycloak, and security token generation. However, to address current challenges of scalability and sustainability, it is necessary to extend this architecture by integrating monitoring tools and indicators that enable more efficient resource management.

This extension of the PoC integrates monitoring systems that provide visibility into the network's status, performance, and energy consumption of distributed resources. The collected indicators allow for bottleneck detection, performance optimization, and improved decision-making. Additionally, mechanisms for automatic resource selection are implemented using indicators and policies that prioritize resources based on parameters such as energy consumption, processing capacity, and other factors, as described in the previous sections.

This involves prioritizing nodes with lower energy consumption or better performance, thus improving the overall efficiency of the federated system. Tests are conducted to evaluate the network's ability to adapt to changes in demand or system conditions, ensuring an ecological and sustainable resource management approach.

In summary, this new phase of the PoC assesses the system's ability to manage resources efficiently, minimizing environmental impact and optimizing consumption and performance in complex federated scenarios.

6.1. Analysis of Practical Experience

The monitor is developed in Python and primarily uses the psutil library. In addition to managing processes, it also provides information about the system and its resources.

The monitor provides a detailed and structured view of the performance and activity of the main system resources. The metrics it collects include essential information about the battery status, such as its current charge level, estimated time remaining, percentage of wear and tear and whether the device is connected to an external power supply.

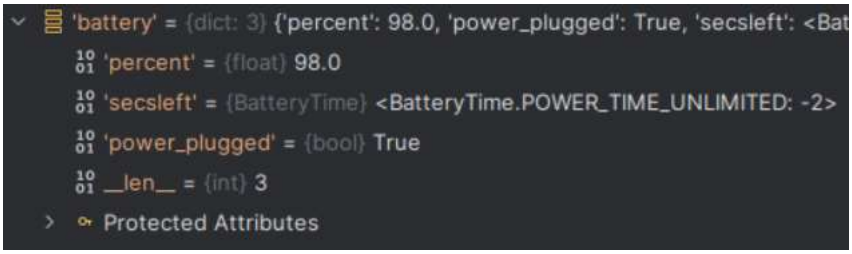

FIGURE 4 BATTERY INFO CAPTURED BY THE MONITOR

This allows us to manage energy consumption efficiently and to anticipate charging needs. For instance, if the drone is low on battery or charging, we will not ask it to deploy a photo service for

land use classification, as it will take time before it can go to its destination. Instead, we will look for another possible cluster.

As for the network, it provides comprehensive data on each available interface, showing the volume of traffic traversing the connection, both in terms of bytes sent and received, as well as the number of packets handled. It also analyzes potential problems, such as errors in data transmission or reception, collisions and statistics related to connection performance, which is essential for diagnosing and resolving network issues. Adding some metrics derived from the previous ones by our own implementation, we have a clear picture of how overloaded the interfaces are, the latency of the network and the use that is being made of them.

FIGURE 5 ENP0S3 INTERFACE STATUS

In the previous image we can see the metrics sent for the enp0s3 interface, which serves as the cluster's external connection in this case. While any interface could be analyzed for internal traffic, our focus here is on assessing the network load. The metrics reveal minimal incoming traffic (The speed is in Mbps) so the cluster would be a suitable candidate for transmitting large amounts of data, such as images or video.

FIGURE 6 ENP0S3 NETWORK CHARACTERISTICS

The monitor also provides static information about the interface itself such as whether it is lifted, whether it is duplex, its speed or its mtu.

Processor performance is another key aspect that the monitor evaluates. It includes detailed metrics on CPU usage times, classifying them by state (active, idle, standby, etc.), in addition to recording the total processor load. Internal behavior is also analyzed, such as context switches between processes and the number of interrupts, which helps to detect bottlenecks and optimize task execution.

FIGURE 7 CPU TIMES DATA

FIGURE 8 CPU STATISTICS AND USAGE

On the storage side, the monitor examines disk activity, providing data on the number of read and write operations performed, the volume of data transferred and the response time of the disks. These statistics make it possible to evaluate storage utilization and detect potential problems, such as slow performance due to excessive access or device errors.

FIGURE 9 TWO ACCESES TO THE DISK DATA

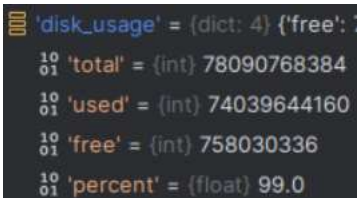

FIGURE 10 DISK USAGE PERCENTAGE

Overall, this monitoring system is invaluable for technical users and administrators, as it not only provides a clear picture of the status and usage of resources but also allows informed decisions to be made to optimize performance, identify potential problems and ensure system stability.

Having seen how the monitor collects system information, we can see how the central node concentrates, organizes and displays it:

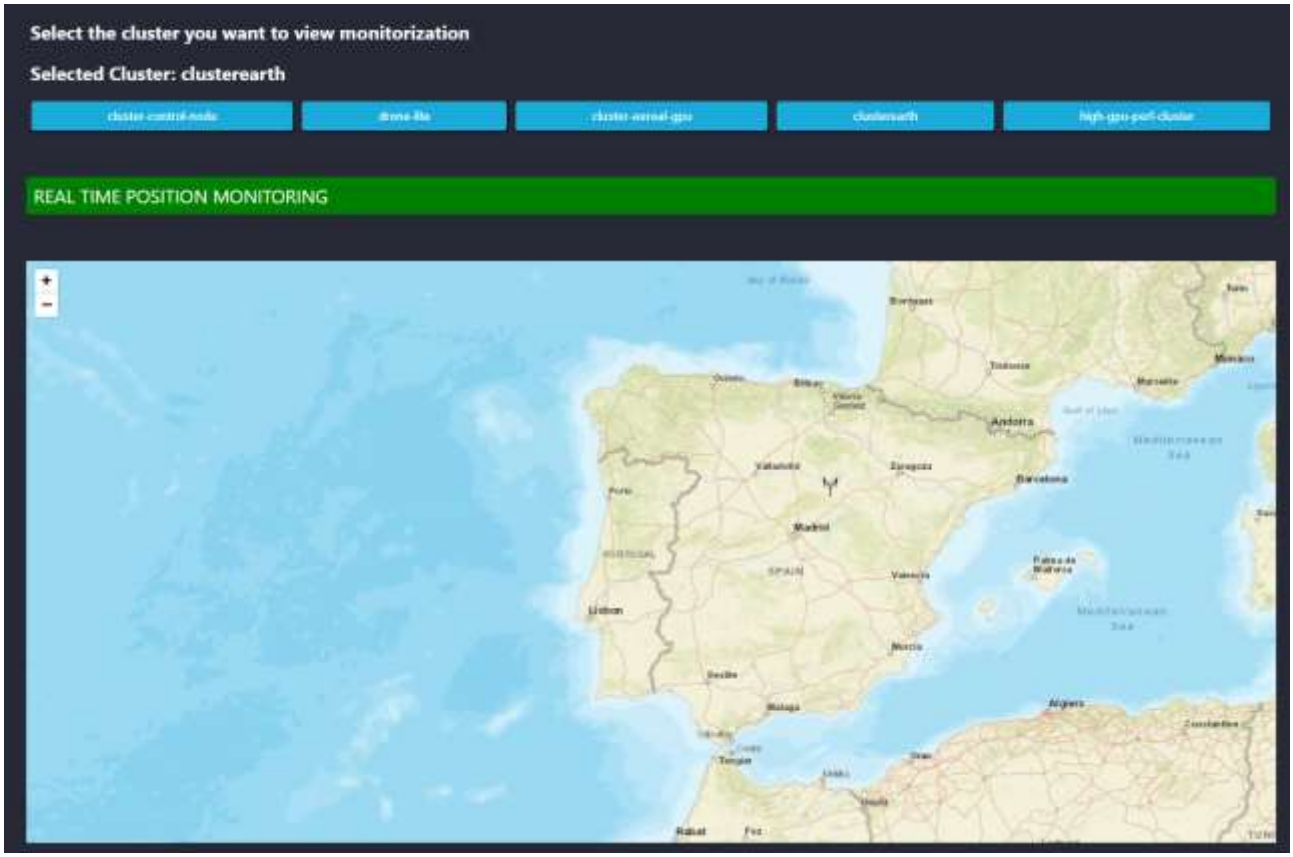

FIGURE 11 CLUSTEREARTH POSITION

The location of the cluster is the first that can be seen on the monitor dashboard. In this case, Clusterearth, as its name suggests, is a fixed-position ground cluster. It is a high-medium power cluster, and typically, these types of machines are stationary and not designed for being mobile.

FIGURE 12 HARDWARE MONITORING AFTER A DOWNLOAD

The main hardware monitoring tells us how the machine is performing and how much work it is handling. We can check mainly the CPU, HDD and RAM statistics. A download was performed to test the monitoring tool, and we can see that the HDD has been in use until we delete the downloaded file.

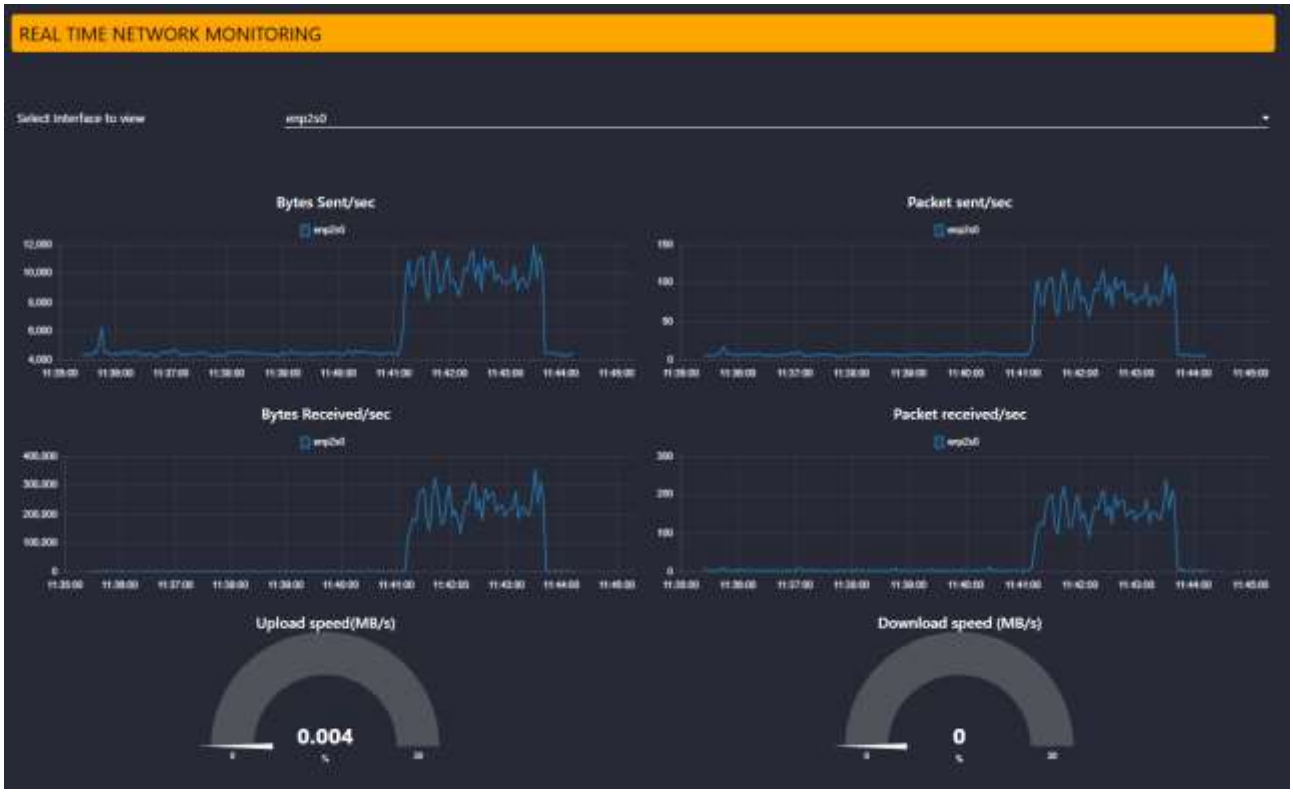

FIGURE 13 NETWORK MONITORING AFTER A DOWNLOAD

The download is reflected on the network monitoring, in which we can select any interface and see the bandwidth available and used. During the time the file was being downloaded, shown in the image under the title as "Bytes received", the download speed, increased. This shows the decision-maker that maybe this cluster is not the best for deploying a service which needs a lot of bandwidth.

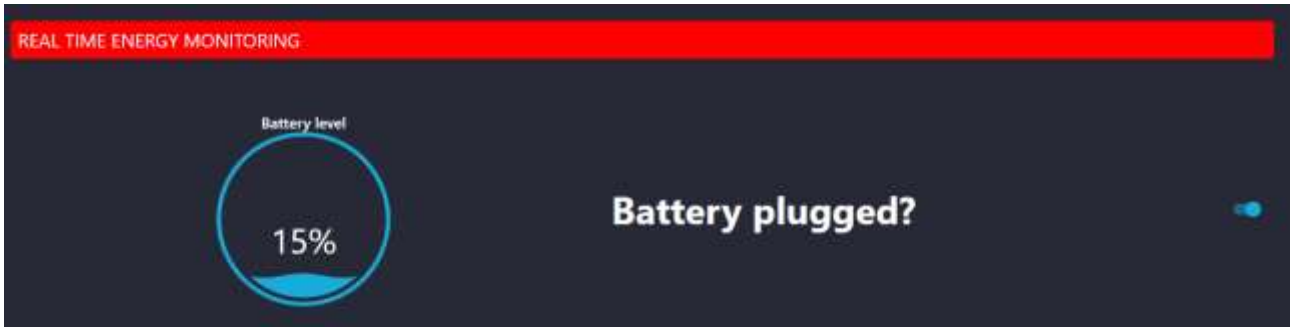

FIGURE 14 ENERGY MONITORING

Finally, for rechargeable clusters, we can monitor the battery. If the battery is charging, the cluster cannot be used. Additionally, if it is known that the battery level is, for instance, below 10% and there is another cluster with the battery almost full, it makes sense to propose the second one cluster to deploy the service.

FIGURE 15 RAM USAGE MONITORING

And for additional examples we can stress the cluster by launching RAM-intensive tests and observing how it is reflected on the data. A cluster with almost not available RAM is not suitable for a high RAM consuming service.

Finally, it is important to remark that all this data is not only displayed but also used at a back-end to propose the user the optimal cluster in which this application can be deployed.

7 Conclusions

By leveraging a simple module containerized within a lightweight and easily deployable Docker setup, the system delivers valuable insights into the machines hosting each monitor. Deployment is straightforward, and data visualization is automated via dashboard, seamlessly integrating an additional layer of information to enhance deployment optimization across federated nodes. Furthermore, it provides fundamental diagnostic data for troubleshooting, adding substantial value to the overall system management process.

If more advanced monitoring capabilities are required, the tools mentioned in this document can be integrated or utilized to complement the system without any collision with the proposed one. These tools offer a deeper level of analysis and enhanced features, ensuring that the monitoring solution can scale and adapt to the evolving demands of any deployment scenario.

8 References

[1] B. Brazil, "Prometheus: Up & Running. O'Reilly Media.," [Online]. Available: https://theswissbay.ch/pdf/Books/Computer%20science/prometheus_upandrunning.pdf. [Accessed 2024].

[2] "Zabbix Documentation," [Online]. Available: https://www.zabbix.com/documentation/. [Accessed 2024].

[3] "Server & Application Monitor documentation," [Online]. Available: https://documentation.solarwinds.com/en/success_center/sam/content/sam_documentation.htm. [Accessed 2024].

[4] "Nagios XI Documentation," [Online]. Available: https://www.nagios.org/documentation/. [Accessed 2024].

[5] "Prometheus documentation," [Online]. Available: https://prometheus.io/docs/introduction/overview/. [Accessed 2024].

[6] "Datadog documentation," [Online]. Available: https://docs.datadoghq.com/all_guides/.