Towards a Functional Continuum Operating System – ICOS MetaOS

Artur Jaworski[1](✉), Marcin Kotlinski[1], Izabela Zrazinska[2], Xavier Masip-Bruin[3], Jordi Garcia[3], and Iman Esfandiyar[4]

[1] Poznan Supercomputing and Networking Center, Jana Pawla II 10st, 61139 Poznan, Poland
arturj@man.poznan.pl
[2] Worldsensing, Viriat 47 10th Floor, 08014 Barcelona, Spain
[3] UPC BarcelonaTECH CRAAX Lab, Rambla Exposicio 59, 08800 Vilanova i la Geltru, Spain
[4] Lukasiewicz Research Network–Poznan Institute of Technology, Ewarysta Estkowskiego 6 st, 61755 Poznan, Poland

Abstract. This article presents the intermediate results of the ICOS project, which aims to create a meta-operating system for cloud continuum, and describes the adaption of ICOS in two (out of four) project's pilot Use Cases. For each described scenario, the paper provides insight into the specific key architectural features, identifies the expected benefits, and provides some initial validation results. The purpose of this research is to highlight the advantages of using a metaOS in real scenarios as well as to illustrate the ease of integration into ICOS.

Keywords: ICOS · Continuum · MetaOS · Internet of Things

1 Introduction

1.1 ICOS as an Implementation of Cloud Continuum

The cloud continuum has been defined as an extension of the traditional cloud towards multiple entities, such as edge, fog, and the IoT, providing analysis, processing, storage, and data generation capabilities [1]. The continuum leverages the combined benefits of both computing paradigms -in short, unlimited resources and performance at the cloud, and low latency, reduced network usage, and increased data privacy and security at the edge.

An effective management of the continuum is a complex task and poses several research challenges that need to be addressed. For instance, managing a large number of heterogeneous, highly dynamic and mobile devices, considering the distributed nature of the different data assets as well as their privacy and security constraints, selecting the appropriate set of nodes where a multi-component application should be deployed for an efficient execution, or considering interoperability and interdependency between the various virtualization technologies of the running nodes, just to name a few.

Resources management and container orchestration at the cluster level is a relatively mature topic with several well-known products on the market, such as Kubernetes [2],

M. Presser et al. (Eds.): GIECS 2024, CCIS 2328, pp. 119–133, 2025.
https://doi.org/10.1007/978-3-031-78572-6_8

Docker Swarm [3] or OpenShift [4]. However, extending the scope of management and orchestration to multi-cluster and heterogeneous edge environments becomes a more challenging topic of current research, which is attracting attention of a large number of researchers, academics and tech developers.

ICOS [5] is a research project that aims to design, develop, and validate a meta-operating system for the Cloud-Edge-IoT continuum by addressing four main challenges:

- device heterogeneity and volatility, continuum infrastructure virtualization and diverse network connectivity,
- optimized and scalable application execution and performance, including resource consumption, guaranteed trust, security and privacy,
- reduction of development and integration costs,
- effective mitigation of cloud provider lock-in effects.

Efforts towards these objectives will be consolidated in a data-driven system built upon the principles of openness, adaptability, data sharing and a future edge market scenario for services and data orchestration.

1.2 ICOS Architecture

ICOS has been conceived as a dynamic and elastic metaOS platform distributed across the continuum. The primary tasks of ICOS are twofold: a detailed management of heterogeneous resources in real-time, and an efficient deployment of applications on multiple nodes along the continuum. The ICOS design has been defined through two different roles: the ICOS Controller and the ICOS Agent. On the one hand, the ICOS Controller is responsible for managing the continuum (tracking the current system topology and availability) and the run-time (deploying and monitoring application execution on demand). On the other hand, the ICOS Agent is responsible for executing the offloaded users' applications, taking care of code execution, data access, telemetry collection and, eventually, runtime communication with other Agents. There is an ICOS Agent running on each node of the continuum (whether a complex cluster or a constrained device at the edge) and it is the only ICOS software running on the remote infrastructure.

ICOS has been designed as a distributed, multi-controller system, in which all relevant decisions are made at the Controller level. ICOS Agents receive instructions from ICOS Controllers and translate them into infrastructure-specific commands. With this organization, ICOS is shaped as a technologically independent platform, where abstract decisions are made without technological restrictions (at the Controller level) and are implemented taking advantage of the capabilities of remote infrastructure technologies (at the Agent level). A typical ICOS scenario is illustrated in Fig. 1.

ICOS Controllers are distributed along the continuum to leverage locality, providing fast response time and supporting scalability. ICOS Agents are deployed at every node of the continuum. They can range from powerful clusters to constrained computing devices and provide processing and data accessing capabilities. Each ICOS Controller is in charge for managing a number of Agents based on locality principles. Upon request for application execution, the Controller will attempt to find locally an appropriate set of nodes within its scope to execute the application according to the application requirements; in case the request cannot be satisfied with the available resources, the Controller

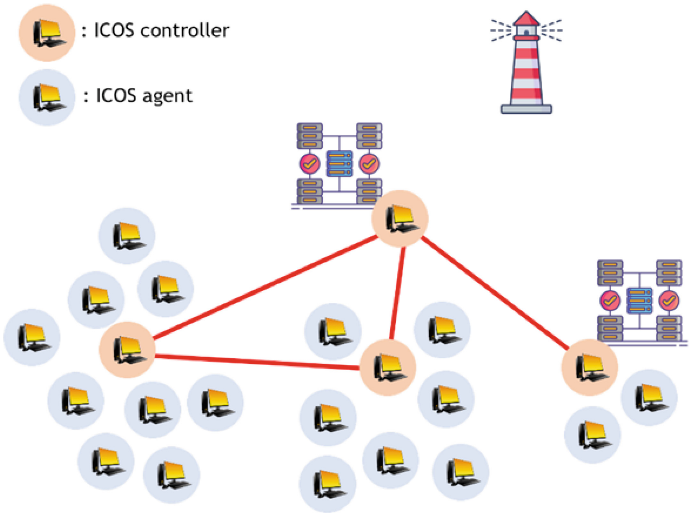

Fig. 1. Typical ICOS scenario.

will coordinate horizontally with other Controllers to find a global solution for the application request.

The ICOS Controller has been designed as a three-layer architecture. As shown in Fig. 2, the Meta-Kernel layer implements all tasks related to the continuum management, as well as the runtime decision making and management. This layer is also responsible for collecting telemetry data from the infrastructure through the Agents. The Intelligence layer is fed by the telemetry data, and it is the responsible for providing intelligence to the Meta-Kernel layer (both for the continuum and runtime management decisions) as well as providing predictive monitoring to forecast different runtime events (resource

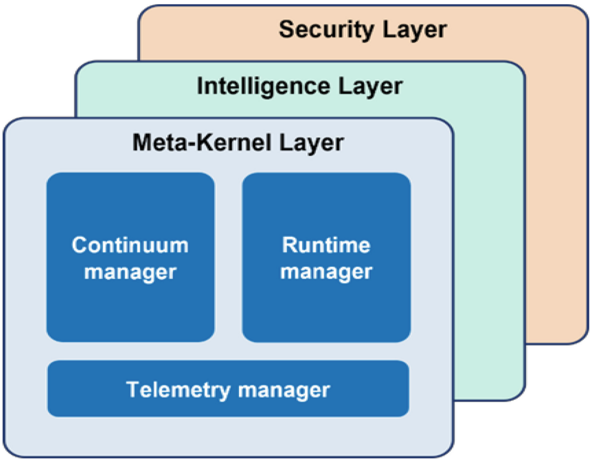

Fig. 2. Architecture of the ICOS Controller.

utilization, network load, or security risks, among others). This layer is also responsible for launching training and retraining processes through federated learning to keep the intelligence model updated. And finally, the Security layer is responsible for providing identity and access management, trusted and encrypted communication channels, as well as detecting security related risks (vulnerabilities, threats, or anomalies) and events (audit).

1.3 Objectives

In this paper, we present the intermediate results of the ICOS project by describing the integration of two real scenarios (as part of the ICOS project's Use Cases [6]) and discussing the challenges in the pilot's integration with the ICOS Alpha release. ICOS has been conceived to face the following main challenges:

- enabling technology agnostic operation in a heterogeneous continuum infrastructure,
- facilitating an on-demand ad-hoc and AI-assisted development of the continuum infrastructure,
- ensuring reliability and security of the continuum,
- creating an open platform facilitating resources, models, data and services sharing, promoting EU innovation and new business models in the continuum arena [7].

The paper is organized as follows. After describing the main architecture and features of ICOS, in Sect. 2 and Sect. 3 we present the two project's pilot use cases, in the areas of agriculture and railway system, respectively. For each pilot, we describe the specific architecture, the expected benefits of using ICOS, the integration with the Alpha release, and the next steps. And finally, Sect. 4 discusses the advantages of using ICOS and concludes the research.

2 Project's Pilot Use Case 1 Perspective

2.1 Overview

In modern agriculture, the seamless flow of data from edge devices to cloud-based platforms is fundamental to maximizing the efficiency and effectiveness of robotic and technological interventions. Distributed processing, data transfer, connectivity, and security are integral components of this data-driven approach [8] (Fig. 3).

At the edge of the agricultural system, sensors, drones, and robotic devices collect vast amounts of data on soil health, crop conditions, weather patterns, and more. These edge devices often operate in remote areas with limited connectivity, necessitating onboard processing capabilities to analyze data in real-time and make immediate decisions. Edge computing minimizes latency and bandwidth requirements by processing data locally, enabling rapid response to changing conditions without relying on constant communication with centralized servers. However local processes are limited by device computational power [9].

The main task of the project at the current stage is to detect weeds and crops and localize them with a centimeter accuracy in global coordinates. This data is then used

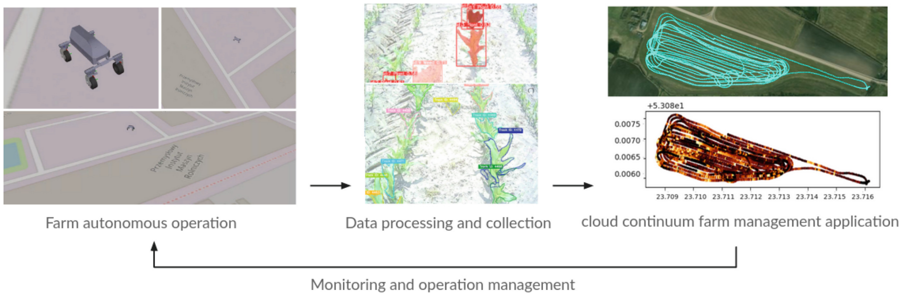

Fig. 3. Use Case's main functionality.

to generate a map of the field containing crop and weed intensity. Once data is collected and preprocessed at the edge, it needs to be transferred to cloud-based platforms for further analysis, storage, and integration with other data sources. High-speed, reliable connectivity solutions, such as cellular networks or satellite communication, facilitate the transmission of data from remote agricultural sites to cloud servers.

In the cloud, agricultural data undergoes extensive analysis using advanced analytics, machine learning, and artificial intelligence algorithms. These tools extract valuable insights from raw data, identifying parameters that inform decision-making processes. For example, predictive analytics models can forecast crop yields, diseases, or optimal planting times based on historical data and current environmental conditions.

Managing the comprehensive system of agricultural data flow from edge to cloud, including data transfer, distributed processing, and storage, poses a complex challenge that often falls beyond the expertise of agricultural software developers. To address this challenge, leveraging an infrastructure capable of supporting such an environment becomes essential. The main challenge to be addressed by integrating with ICOS is providing continuous robot operation in rural areas with low network coverage and high latency.

2.2 Architecture

Use Case's architecture spreads across cloud, edge and IoT domains. Depending on the mission, the robot can take advantage of cloud computing, for instance, detecting and localizing weeds and crops based on AI models. The data generated during the mission, along with the status and record of the operation in the field, is sent to cloud infrastructure. The cloud processes analyze the farm condition and robot health and generate a farm yield map. This generated data is then sent to the robot as the next task and mission to be executed on the farm. For instance, the robot might execute a spraying mission at the part of the farm where weak crop conditions were previously detected.

Hardware Specification. The robotic platform, shown in Fig. 4, has mechanical tools for autonomous seeding and spraying, equipped with a hydraulic power drive system and a diesel engine. The robot takes advantage of four active steerable wheels equipped with an independent suspension system for a smooth maneuver on the farm (Fig. 5).

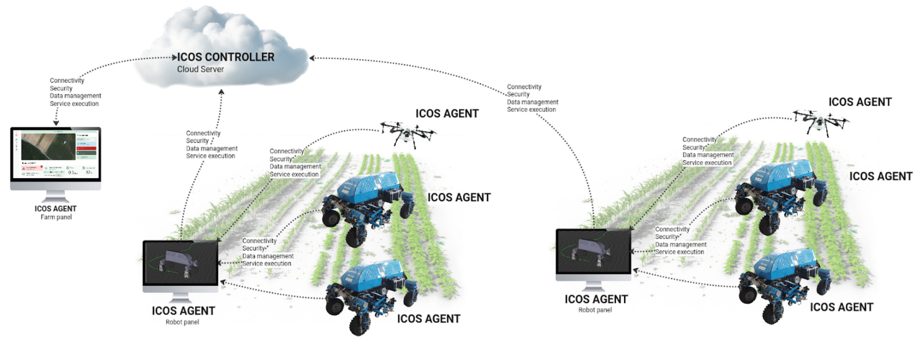

Fig. 4. Use case architecture overview.

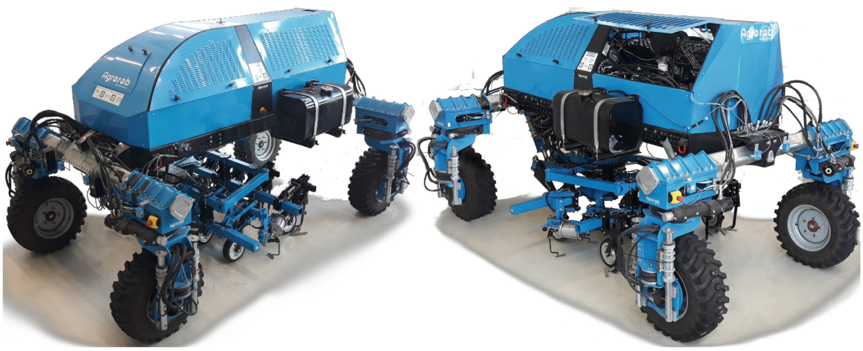

Fig. 5. AgroRob, the agricultural robot used in this work.

A 24 V designated battery pack connected to the alternator of the diesel engine, powers the high-performance onboard computer. The robot's CAN bus is connected to this computer via USB adapter. To establish connectivity, an industrial 5G router is connected via LAN port.

The robot is localized by GNSS receivers, as well as odometry data received from wheels' encoders [10]. Three RGB cameras and a depth camera are mounted on the robot as the vision sensors. Robot Operating System version 2 (ROS2) [11] has been used as an interface between robot and the computer, while You Only Look Once, version 8 (YOLO8) [12] is responsible for image recognition.

2.3 Expected Benefits

End User added value for the Use Case implementation:

- enhanced operational efficiency: autonomous robots perform precise tasks such as seeding, weeding, and spraying, reducing the need for manual labor and increasing task accuracy,
- cost reduction: predictive maintenance and optimized resource application lead to significant savings on fertilizers, herbicides, and repair costs,

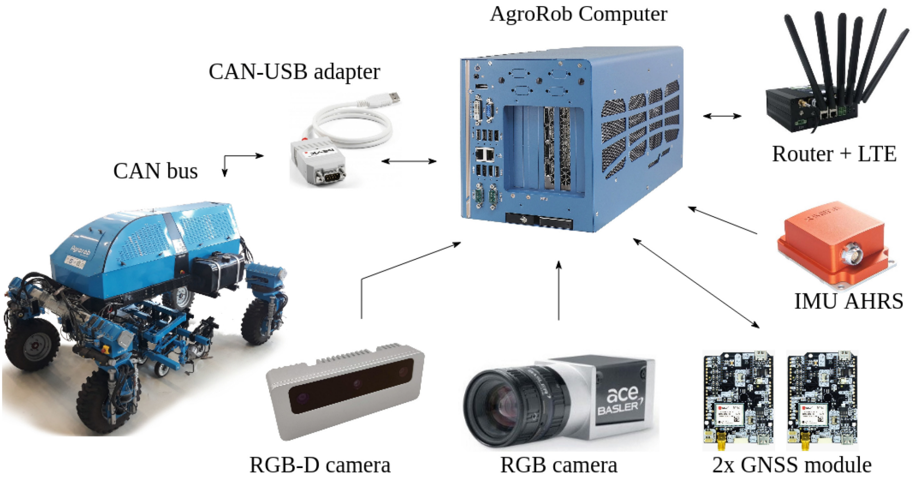

Fig. 6. Hardware setup, including communication hardware and sensors

- sustainability: efficient resource use and reduced chemical application promote environmentally friendly farming practices,
- improved decision-making: real-time data and analytics support informed decisions, optimizing crop management and increasing yield.

Applications Executed with ICOS Support and Requirements (Challenges) for ICOS:

- Autonomous Field Monitoring: this application utilizes the robot to autonomously monitor crop conditions, soil health, and environmental factors. The platform, equipped with advanced sensors and cameras, collects real-time data to enable precise interventions, enhancing crop health and yield. The application requires ensuring reliable integration of various sensors, managing large volumes of data, and providing actionable insights in real-time. There is also a need for robust connectivity solutions such as LoRa [13], WIFI, and xG modems to facilitate data transmission.
- Weed and Disease Detection: leveraging machine learning and computer vision, this application identifies weeds and diseases early, enabling targeted treatments. The system processes data locally on the edge to ensure timely responses even with intermittent connectivity. The weed map generated during the robot's first pass helps in precision treatment during subsequent passes. The application requires developing accurate detection algorithms, managing the computational load for processing image data on the edge, and maintaining high performance under varying field conditions.
- Predictive Maintenance and Resource Optimization: this application uses predictive analytics to anticipate machinery maintenance needs and optimize resource usage. By analyzing sensor data and operational logs, it schedules maintenance proactively and adjusts resource application to prevent overuse. Data from cameras, logs, and other devices are stored in the cloud, with predictive analysis considering vibrations and control signals. The application requires creating robust predictive models, balancing computational loads between edge and cloud resources, ensuring data security, and

developing user interfaces for maintenance management and parameter control. There is also a need for securing the connection due to the high value of robotic devices.

2.4 Integration with the ICOS Alpha Release

Once the requisite hardware has been installed, the iterative implementation of ICOS could commence. As all the operating systems used in Use Case support Docker, the next step was to select an orchestrator. Nuvla [14] has been selected for this purpose. Nuvla serves as a comprehensive platform that facilitates the seamless deployment, monitoring and scaling of "dockerized" applications, thereby increasing the efficiency and reliability of containerized environments.

With all these preparations completed, the actual onboarding could begin. A virtual machine, located on PSNC premises and hosting the Zenoh router, was selected for initial onboarding. The Edge component was registered in Nuvla and subsequently deployed on the Docker engine running on the virtual machine, utilizing the provided docker-compose YAML files. Following the completion of the deployment, the Edge registered in Nuvla underwent a transition to the operational status within a few seconds. The procedure was executed without incident, in accordance with the instructions provided in the documentation.

The final stage of the process involved the deployment of the ICOS Telemetry Agent on the virtual machine that had been previously onboarded and the subsequent connection of this agent to the Telemetry Controller, which was running on the staging testbed for Use Cases integration. This final step may be divided into two sub-steps: firstly, establishing a connection to the testbed via a VPN client, and secondly, initiating the Telemetry Agent. VPN client was deployed on the virtual machine, thereby establishing a connection to the testbed. Following the implementation of a series of corrective measures, the deployment was successfully concluded, resulting in the commencement of data reporting by the Telemetry Controller, as evidenced by the Fig. 6. Both the VPN client and the Telemetry client have been deployed by Nuvla (Fig. 7).

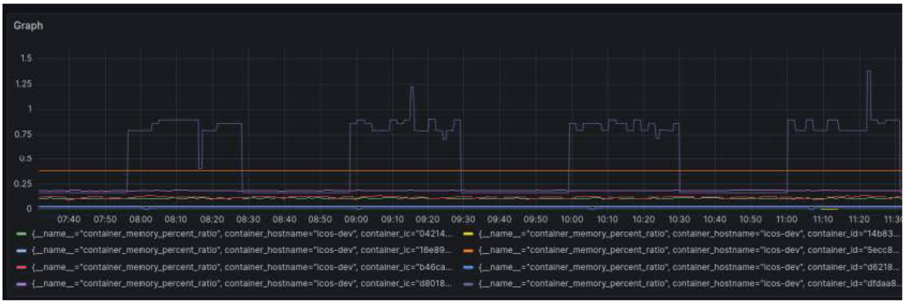

Fig. 7. Telemetry data from UC1 virtual machine being reported by the Telemetry Controller.

2.5 Next Steps

The subsequent stages of integration into the ICOS environment will involve preparation for the Beta release. The main point of interest for the Use Case will be validation of own applications deployment using ICOS on both the cloud and the robot. For this to happen, Docker Compose YAML files will have to be translated into ICOS Application manifest format.

After successful deployment, the work on leveraging ICOS's Intelligence Layer will commence. It is expected to be a base for creating the aforementioned Predictive Maintenance module, as well as help in training more accurate image recognition models used on the robot.

Apart from work directly connected to using ICOS, other modules will also be revamped. The team will focus on creating user dashboards for web and mobile devices. These applications will allow users to plan missions, see their results, as well as monitor the robot's parameters in real-time.

3 Project's Pilot Use Case 2 Perspective

3.1 Overview

Ferrocarrils de la Generalitat de Catalunya (FGC) infrastructure includes metro and commuter lines in and around the city of Barcelona, tourist mountain railways, and rural railway lines which serve more than 90 million passengers per year. Whilst most lines are conventional adhesion railways, the FGC also operates two rack railways and four funicular railways.

On all these railway lines, the massive deployment of sensors along different parts of the infrastructure is essential for the optimization and improvement of service and safety. The increasing number of sensors and their specific, and typically siloed solutions, present an increasing complexity related to the management and operations of such solutions.

Today, the railway monitoring process to improve the maintenance cycle is basic, and for most railway operators it is done preventively (once every fixed period) through a special train with sensors which runs through the whole rail system [15]. This special train can measure several key parameters of the railway system, such as the height difference and width between the rails, and thus identify where, potentially, corrections in the track geometry is needed. However, this measurement is only taken once or twice a year; in the remaining months, nobody knows what happens (only physical inspections are available: very costly and uncommon), and there is no established procedure to evaluate the cost-effectiveness of the actions taken to address the identified rail tracks' issues. Indeed, digital technology, such as IoT, aims to minimize the monitoring and maintenance costs by gaining knowledge of the status of key aspects of the railway infrastructure in real-time: rail tracks geometry, slope, surrounding areas settlements and falling elements, overhead lines maintenance, etc.

The main challenge to be addressed by the Use Case is related to the continuous monitoring of critical infrastructure on rail tracks to ensure safety and improve maintenance activities.

The initial area to deploy and validate the Use Case for the Railway Structural Alert Monitoring system (RSAM) is the line in Lleida-La Pobla due to its difficult access to several of the areas of the line and its orography generating possible geological incidents (Fig. 8).

Fig. 8. Safety problem on the FGC rail track.

3.2 Architecture

In November 2023 (M15), as part of Use Case 2, a multitude of IoT devices were strategically deployed along the FGC rail tracks. The deployment site spans the Lleida-La Pobla line, covering a 4 km stretch within the challenging terrain of Gerb. This particular rail line, facilitating 16 train circulations daily, stands as the sole transportation artery for the region. Characterized by its precarious geodesic conditions, including water flow beneath the tracks and non-compacted layers, as well as the presence of small caves and historical instances of ground collapse, the area underscores the critical need for a real-time monitoring solution. This solution, intended to significantly impact end-users, notably the railway operator FGC, is being implemented by Worldsensing (WSE) within this Use Case and is set for validation by ICOS (Fig. 9).

Fig. 9. IoT devices installed on the FGB railway track as a part of ICOS Railway Use Case

For the Railway Structural Alert Monitoring system, ICOS will be managing the Edge and Cloud processing environments. Edge will be supported by the IoT Gateway with limited resources for computing and 4G connectivity through commercial mobile services to the Cloud computing environment. The cloud computing environment used by Worldsensing is provided by Google Cloud Platform. Both the Edge device and the Cloud environment should have the ICOS agent deployed to be able to onboard such elements to the continuum.

The onboarding of both compute services will allow the orchestration of services through ICOS Meta OS according to specific requirements for the Monitoring, Safety and Maintenance applications available in the CMT Cloud solution.

Out of the scope of the ICOS-managed environment, data from the IoT sensors and nodes will be aggregated at the IoT gateway through LoRaWAN radio communication. The data collected from the IoT sensors (tiltmeters) is related to the geometry parameters of the rail track, while IoT nodes collect data from geotechnical sensors (extensometers and piezometers) to geological parameters (Fig. 10).

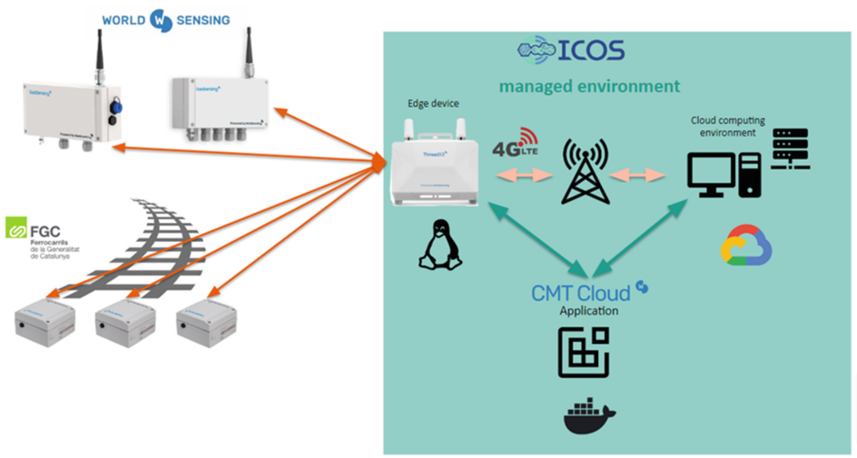

Fig. 10. Use Case 2 architecture.

3.3 Expected Benefits

End User added value for the Use Case implementation:

- timesaving through continuous monitoring: limit intensive personnel inspections that are done every day before train circulations,
- cost saving: implement corrective actions in advance to avoid reparation costs,
- improve safety: establish velocity limits to avoid risky situations when with quality of the operation decreases.

Applications executed with ICOS support and requirements (challenges) for ICOS:

- Real-time Monitoring: the application will support safe operations by deploying a digital and wireless monitoring system that will collect and deliver real-time information regarding the quality parameters to monitor critical infrastructure status to support the decision makers and to timely detect possible anomalies or physical threats regarding the railway track.

- Critical event detection for Safety: the alarm detection module is connected to the deployed devices, and it allows the detection and acquisition of possible alarms. The detection of alarms and response actions can be required to be processed at the edge to ensure safe operations even if connectivity with upper layers is not fully available. Such response to events detected might also include the request for additional information to the physical devices to collect additional contextual data about the possible incidents and thus to better design or select a response plan.
- Prediction for maintenance planning: to optimize the decision-making process and exploit all the available resources, the maintenance application will also be onboarded within the ICOS architecture. The objective of the application is to identify the trend and predict the moment when the condition where quality parameters would not be met, and therefore plan maintenance activities to mitigate such risk. The proposed application will request available resources at the edge and cloud level based on connectivity and data transfer requirements and will run appropriately.

3.4 Integration with the ICOS Alpha Release

To prepare the ICOS for its initial deployment, the WSE team tested different Gateway models, one of which is currently under development.

Deploying Nuvla Agent on Gateway. The objective of the test was to be able to deploy a Nuvla agent in a Gateway, using Docker. The WSE team tried to keep track of the CPU and memory usage in the Gateway, to install more services (such as Telemetry or VPN Agent) in the future, the team has chosen to use htop, to monitor the state of the system. The use of a CPU with components that are needed for basic Gateway functionalities uses around 50% of its capacity. Once the team started the docker services, there was a

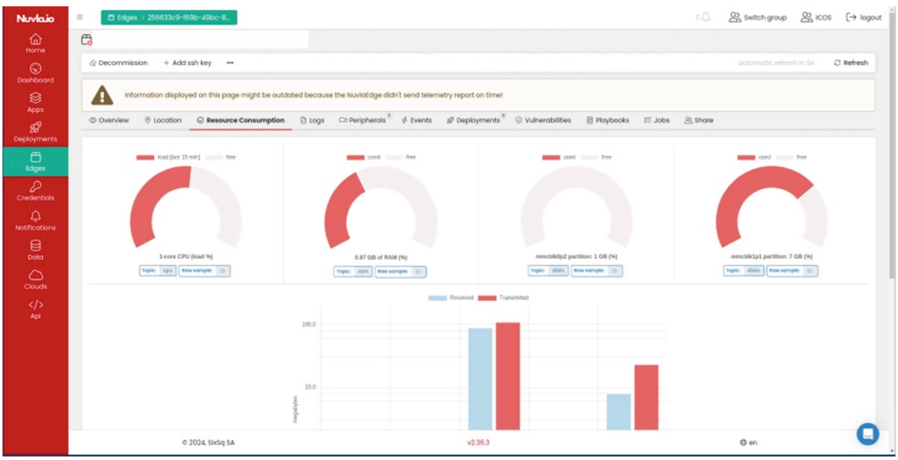

Fig. 11. Nuvla dashboard for UC2 resource consumption on the Worldsensing Gateway.

noticeable increase in CPU but the memory usage did increase, at least with a memory reservation. When the docker-compose images started to work the CPU usage increased clearly (>40%) (Fig. 11).

Based on this test, it was decided to deploy the Nuvla agent using Docker, instead of Kubernetes, as Kubernetes could not be configured satisfactorily and the CPU usage increased more than with Docker.

Deploying Telemetry and VPN Agents via Nuvla. The final step was to deploy Telemetry and VPN Agents remotely. It allowed to confirm that the device will be able to maintain its functionality. In further steps, some improvements will be made to maintain the GW and the system stable.

3.5 Next Steps

The next steps for the Use Case 2 ICOS setup involve collecting and adapting docker-composed architecture (Nuvla Agent) for edge application components installed on the Gateway and translating Kubernetes (k8s) manifests for cloud application components (CMT). These manifests will first be adapted to the ICOS manifesto template. Following successful testing, the deployment will be conducted through ICOS, with careful monitoring to ensure all components are operational. In later stages, UC2 might consider testing ICOS implementation with the OCM agent which could be installed and tested to verify its interactions with the cloud components.

Once Use Case 2 is fully onboarded into the ICOS system it is expected to focus on functionalities that will bring the impact to end user operations such as:

- real-time monitoring: data integrity and synchronization if there are connectivity problems between Edge-Cloud,
- critical event detection for safety: operate regardless of connectivity (taking local decisions),
- prediction for maintenance planning: identify the trend and predict the moment when quality parameters would not be met.

By the end of the project, we aim to confirm the business value that the use of the ICOS system brings to transportation. This includes enhanced safety and efficiency due to real-time data collection and edge processing, which allow for immediate response to potential issues. The system facilitates cost savings through predictive maintenance and optimized resource allocation, while also improving passenger satisfaction by reducing delays and enhancing service reliability. Compliance with regulatory standards is maintained through precise, up-to-date data, and advanced asset management is enabled by comprehensive cloud analytics, optimizing railway operations and infrastructure management.

4 Conclusions

As presented, ICOS strives to create an easy to integrate solution, which will enable usage of the edge-to-cloud continuum by its adopters. All the modules, which compose the architecture, as well as their features, are iteratively consulted with the early adopters

- four project's pilot Use Cases (two of which were presented in this document), as well as organizations selected in two Open Calls (the second one is open for applications at the time of writing this document). Choosing ICOS architecture as a base for developed solutions will help minimize their time to market, improve reliability and open new possibilities for maximizing benefits of using cloud, edge and IoT domains in a form of a continuum. Our future efforts will be focused on validating the new features and providing feedback during the ICOS development.

Acknowledgments. The ICOS project has received funding from the European Union's HORIZON research and innovation programme under grant agreement No 101070177.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Moreschini, S., Pecorelli, F., Li, X., Naz, S., Hästbacka, D., Taibi, D.: Cloud continuum: the definition. IEEE Access **10**, 131876–131886 (2022)
2. Kubernetes Homepage. https://kubernetes.io. Accessed 20 June 2024
3. Docker Swarm Homepage. https://docs.docker.com/engine/swarm. Accessed 20 June 2024
4. OpenShift Homepage. https://www.redhat.com/es/technologies/cloud-computing/openshift. Accessed 20 June 2024
5. ICOS Homepage. https://www.icos-project.eu/. Accessed 20 June 2024
6. Zrazinska, I.: ICOS. Deliverable 6.4: Use Cases settings and demonstration strategy (2023)
7. D'Andria, F.: ICOS. Deliverable 2.1: ICOS ecosystem: technologies, requirements and state of the art (2023)
8. Debauche, O., Mahmoudi, S., Manneback, P., Lebeau, F.: Cloud and distributed architectures for data management in agriculture 4.0: review and future trends. J. King Saud Univ. - Comput. Inf. Sci. **34**(10), 9622–9643 (2022)
9. Tzounis, A., Katsoulas, N., Bartzanas, T., Kittas, C.: Internet of Things in agriculture, recent advances and future challenges. Biosys. Eng. **164**, 31–48 (2017)
10. Esfandiyar, I., Ćwian, K., Nowicki, M.R., Skrzypczyński, P.: GNSS-based driver assistance for charging electric city buses: implementation and lessons learned from field testing. Remote Sens. **15**, 2938 (2023)
11. ROS Homepage. https://www.ros.org/. Accessed 20 June 2024
12. YOLO8 Homepage. https://yolov8.com. Accessed 20 June 2024
13. Mikhaylov, K., Petäjäjärvi, J., Hänninen, T.: Analysis of capacity and scalability of the LoRa low power wide area network technology. In: European Wireless, pp. 119–124 (2016)
14. Nuvla Homepage. https://nuvla.io. Accessed 20 June 2024
15. Budai, G., Huisman, D., Dekker, R.: Scheduling preventive railway maintenance activities. J. Oper. Res. Soc. **57**(9), 1035–1044 (2006)