Ref. Ares(2025)7796984 - 17/09/2025

D6.11 Final ICOS Product release

Document Identification			
Status	Final	**Due Date**	31/08/2025
Version	1.0	**Submission Date**	17/09/2025

Related WP	WP6	Document Reference	D6.11
Related Deliverable(s)	D6.4, D6.5, D6.9, D5.3, D2.2, D2.4	Dissemination Level (*)	PU
Lead Participant	WSE	Lead Author	Izabela Zrazinska
Contributors	WSE, SSEA, UniTO, ENG, L-PIT, CRF, ATOS, BSC, FBC	Reviewers	Francesco D'Andria (ATOS)
			Ivan Paez (ZSCALE)

Keywords:
Use Cases, Final release, KPIs Validation, End Users Feedback

Document Information

List of Contributors	
Name	Partner
Artur Jaworski	PSNC
Marcin Kotliński	PSNC
Izabela Zrazinska	WSE
Lee Waddell	SSEA
Gabriele Giammatteo	ENG
Francesc Lordan	BSC
Iman Esfandiyar	L-PIT
Marta Łakomiak	L-PIT
Matteo Miceli	UNITO
Laurentiu Jr Marius Zaharia	UNITO
Fabio Ciravegna	UNITO
Marco Marchetti	CRF
Rosa Villaronga	FBC

Document History			
Version	Date	Change editors	Changes
0.1	06/06/2025	WSE	Document creation. Table of contents
0.2	26/08/2025	PSNC, L-PIT, WSE, UNITO , SSEA, ENG, CRF	Inputs from partners
0.3	05/09/2025	WSE	Final draft
0.4	09/09/2025	ATOS, ZETASCALE	Quality review
1.0	17/09/2025	ATOS	Final version submitted

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	Izabela Zrazinska (WSE)	05/09/2025
Quality manager	Carmen San Román (ATOS)	17/09/2025
Project Coordinator	Francesco D'Andria (ATOS)	17/09/2025

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
AORP	Agriculture Operational Robotic Platform (Use Case 1)
CMD	Command Prompt
CPU	Central Processing Unit
DDS	Data Distribution Service
Dx.y	Deliverable number y, belonging to WP number x
EC	European Commission
EMDS	Energy Management and Decision Support system (Use Case 4)
OC	Open Call
FSTP	Funding Support to Third Parties (Open Call Use Cases)
GA	Grant Agreement
GPS	Global Position System
HMI	Human-Machine Interface
IAIMM	In-car Advanced Infotainment and Multimedia Management system (Use Case 3)
IoT	Internet of Things
IT	Iteration
KPI	Key Performance Indicator
LoRa	Long Range
MQTT	Message Queuing Telemetry Transport
OCM	Open Cluster Management
OS	Operating System
QoE	Quality of Experience
ROS2	Robot Operating System, version 2
RSAM	Railway Structural Alert Monitoring system (Use Case 2)
SW	Software
UC	Use Case
VM	Virtual Machine
WP	Work Package
WPL	Work Package Leader
GW	Gateway
LLM	Large Language Model
DNS	Domain Name System
ISP	Internet Service Provider
CLI	Command line interface
DPM	Dynamic Policy Manager
GUI	Graphical user interface

Executive Summary

This deliverable presents the validation results of final release of the ICOS Meta Operating System, incorporating enhancements driven by real-world validation and feedback from all Use Cases engaged in the ICOS project. It marks the culmination of development, integration, and iterative improvements made throughout the project's life cycle.

The document consolidates the validation results from four core internal Use Cases—spanning agriculture, railway monitoring, automotive, and energy management—as well as 5 + 15 external Use Cases selected through 2 Open Calls.

As part of this document, each core Use Case is accompanied by a demonstration video (see Section 7) that illustrates the added value ICOS delivers within its respective vertical. These demonstrations validate key functionalities in practice and are complemented by a review of Key Performance Indicators (KPIs) from a business perspective.

An important role of the Use Cases within the project was to provide the ICOS team with both structured validation data and qualitative user feedback on functionality, usability, and integration. This feedback loop, documented in Section 4 of this report, ensured that platform development was guided by real-world requirements across multiple domains. The final version of the ICOS platform reflects substantial improvements derived from this cross-sectoral input. These enhancements—covering architectural refinements, upgraded components, and deployment options—are presented in detail in Section 5, which provides the final platform overview.

Overall, the document provides strong evidence that ICOS delivers technological value, business relevance, and sector-specific impact, positioning it as a ready-to-adopt solution for future industrial applications.

1 Introduction

1.1 Purpose of the document

The purpose of this document is to present the improvements made into the final release of the ICOS based on the feedback gathered through validation activities conducted across the project's core and Open Call Use Cases. It aims to demonstrate the ICOS added value and how it was tested in real-world conditions (DEMO videos included in Section 7), how it meets predefined business KPIs (Section 3), and how user and stakeholder feedback has shaped the final improvements (Section 4 and 5).

1.2 Relation to other project work

This deliverable is part of WP6 that is focused on ICOS deployment, validation, demo strategy and final ICOS Platform release, as presented in Figure 1. Therefore, the main goal of this document is to present the improvements made in technical work packages in ICOS development based on end users feedback and validating ICOS functionalities. Use Cases played a key role in exploitation being the first early adopters. They were able to show the added value of the final project results in a real environment and the benefits that they bring to potential future clients.

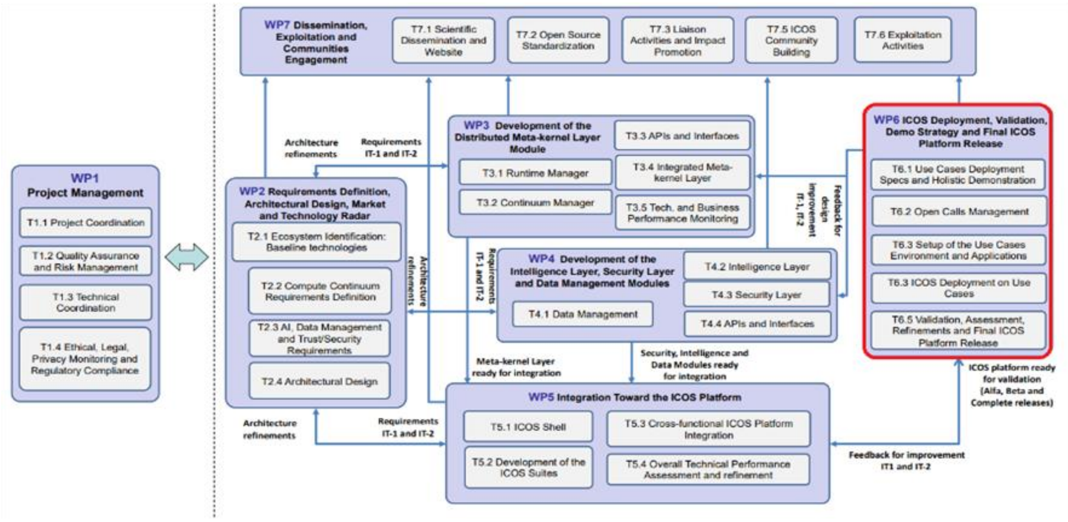

Figure 1 ICOS PERT Chart: WP6 relation to project work

1.3 Structure of the document

This document is structured into six major sections:

- Section 1-2 introduce the purpose of the deliverable and provide an overview of the results achieved within Work Package 6.
- Section 3 provides an introduction for demo of each of four core Use Cases, including business KPIs validation.
- Section 4: provides an overview of all Use Cases feedback about ICOS.
- Section 5 presents a summary of the improvements made to the ICOS platform based on the feedback received from Use Cases.
- Section 6 concludes the document with a summary of achievements and outlines the next potential steps beyond the project duration.
- Section 7 includes links to core Use Cases DEMO videos and references to other documents.

2 Final results of Work Package 6

The objective of Work Package 6 (WP6) was to coordinate the deployment, validation, and demonstration activities of the ICOS solution, culminating in the delivery of the final ICOS platform release. During the ICOS project duration, WP6 focused on the development and implementation of the four core internal Use Cases, the launch and management of two Open Calls —following European Commission standards—to select external use cases: five development projects and fifteen testing and validation initiatives, ensuring a wide outreach and effective management of Financial Support to Third Parties. It also covered the deployment and validation of the ICOS platform across all use cases, both internal and external, with the goal of demonstrating the platform's benefits against defined business and technical KPIs. Ultimately, feedback gathered from Use Cases under WP6 led to the delivery of the final, fully validated ICOS platform.

Table 1 WP6 Objectives and KPIs achieved by M36

Objective	KPI related	Deliverable	Status by M36
O6.1: Development of the four envisioned use cases	KPI 4.3. Specification of use cases and pilots	D6.4 Use cases settings and demonstration Strategy (M15)	Completed
O6.2: Launch and manage 2 Open Calls	KPI 4.6 Successful deployment of the Open calls process	D6.1 Call Announcement and Guide for Applicants - First Open Call (M11) D6.2 Public Summary Report – 1st Open Call (M17) D6.3 First Open Call evaluation report (M18) D6.6 Call Announcement and Guide for Applicants - Second Open Call (M22) D6.7 Public Summary Report - Second Open Call (M29) D6.8 Second Open call evaluation report (M30) D6.10 FSTP payments report (M36)	Completed
O6.3: Deploy and validate the ICOS platform in all use cases (ICOS use cases and Open Calls projects) **O6.4: Demonstrate the ICOS platform benefits vs target KPIs**	KPI 4.4 ICOS validated in 9 project verticals (4 UCs + 5 UCs from 1st open call) KPI 4.5 ICOS validated in at least 15 services (2nd open call) related to the project verticals	D6.5 Report on Validation results (M18) D6.9 Report on Validation results (M34) D6.10 FSTP payments report (M36)	Completed
O6.5: Produce the final release of the ICOS platform	KPI 4.7 ICOS Final release	D6.11 ICOS Final release (M36)	Completed

As presented in Table 1 and within the scope of WP6, all targets related to Objective 4 and Objective 5 have been successfully achieved.

For Objective 4: *Demonstrate the project outcomes in key relevant scenarios*, the ICOS operating system was demonstrated and validated through deployment in the **four core project Use Cases** as well as the **five Use Cases from the first open call**, thereby **covering nine distinct verticals**.

- Four Internal Core Use Cases: agriculture, energy, railway, automotive
- Five Solution Development Use Cases (First Open Call): structural health monitoring, intelligent traffic management, energy management - smart grids, climate and space weather, occupational safety

In addition, validation was extended to **fourteen Uptake Use Cases selected through the Second Open Call**, listed below:

- Smart, real-time water damage detection and prevention
- Environmental Factor-Based Decision System for Smart Cities
- An occupational safety management platform to enhance situational awareness and support decision-making throughout the construction process.
- Structural Health Analytics for Resilient Civil Infrastructure
- EdgeAI for Virtual Distributed Powerplant Optimization
- Optimising home comfort through smart grid integration and intelligent energy consumption
- Smart PPE Compliance for Occupational Safety
- AI-powered UAV solution for detailed follow-up inspections
- Flexible Ai-driven Nilm-based energy management Application through domestic metaOS
- Empowering Homes with AI for Smarter, Greener Energy Solutions
- IoT-Edge-Cloud service to rapidly modulate flexible electrical loads.
- Automatic detection of critical faults in railway catenary.
- Railway safety with real-time structural monitoring using satellite data and AI, providing predictive alerts e2cloud.
- Improving rail safety and efficiency with high-precision train monitoring and secure, immutable blockchain-based data certification.

The final ICOS release delivered in March 2025 (D5.3) was validated by all use cases (KPI4.3). ICOS was validated across nine project verticals (KPI4.4) and forteen services (KPI4.5), the open calls process was successfully executed (KPI4.6), and the ICOS final release was delivered at M36 (KPI4.7).

Figure 2 ICOS Open Calls results

In parallel, Objective 5 *Building an open innovation environment and fostering the creation of new applications in the continuum as well as the science and engineering community* has also been fully accomplished under WP6. The project demonstrated measurable progress on non-functional (business) requirements derived from the use cases such as improving service availability, reduction in delay and reduction in exposure to cybersecurity threats. Details about Business KPIs achieved in each core use case are presented in Section3 of this document. These achievements confirm the effectiveness of WP6 in demonstrating and validating the ICOS outcomes, ensuring both technical performance and alignment with end-user business needs.

3 Use Case Validation Results

This section introduces results of four core use cases, focusing on the validation of ICOS functionalities and the business KPIs achieved. Links to the videos published on the ICOS channel can be found in the Section 7 - References.

3.1 UC1 Agriculture Operational Robotic Platform [PSNC, L-PIT]

3.1.1 Functionality validated (DEMO)

<u>Data synchronization (Eclipse Zenoh & ROS)</u>

Reliability between the Robot and User Control panel is managed by the DDS layer, with reliability determined by the QoS settings applied to message topics at both the data-writer and data-reader levels. In UC1, we employ Zenoh bridges to facilitate communication between the ROS2 environment and Zenoh. Zenoh's built-in capabilities ensure publication catching and subscriber queries for all topics configured with Transient Local durability.

As a key technology leveraged by the ICOS project, Zenoh fully handles data synchronization in UC1. Even after prolonged network disconnections, Zenoh bridges automatically rediscover each other and forward all locally cached messages to the counterpart Zenoh component. By implementing Zenoh in UC1—an agrorobotics demonstration within the ICOS framework, we effectively validated the broader ICOS platform, demonstrating its efficiency in managing distributed, heterogeneous systems across the cloud-to-thing continuum.

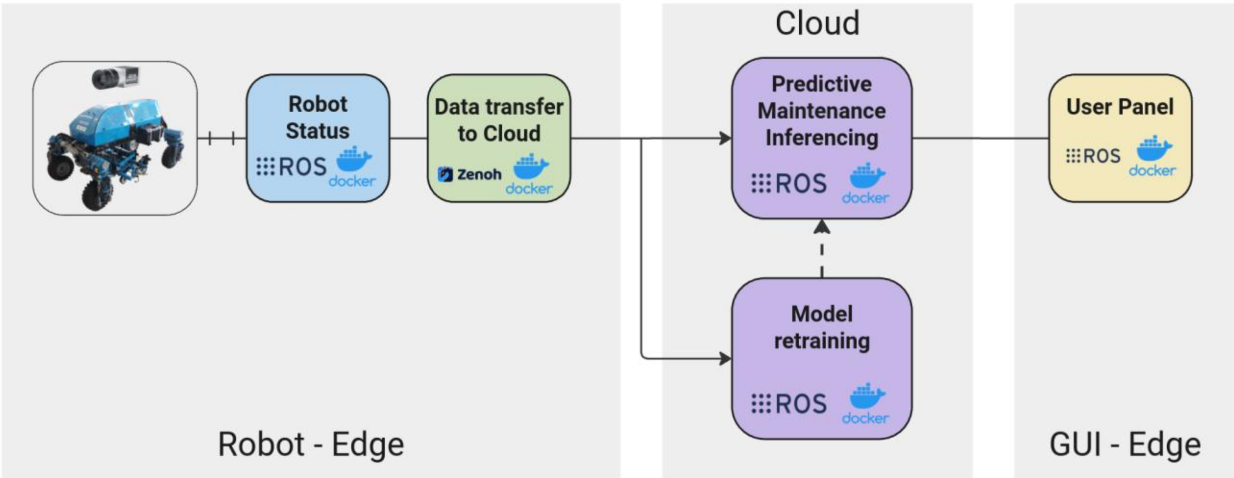

Figure 3 UC1 predictive maintenance architecture

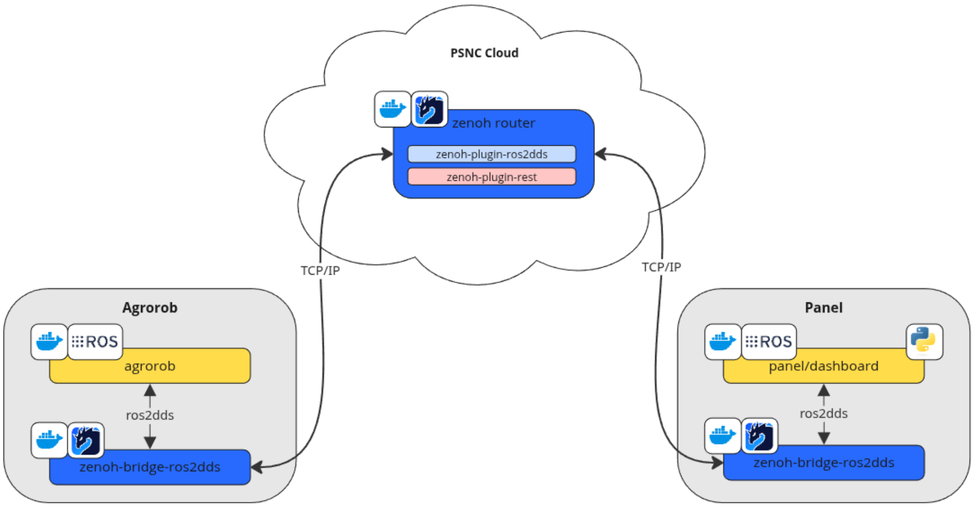

Figure 4 UC1 communication architecture

Intelligent deployment (Modules in app)

Use Case's applications have been split into logical modules and dockerized, to allow their deployment throughout the continuum. At the time of writing, the following modules have been identified:

- Camera driver - used to obtain stream from AgroRob cameras,
- Zenoh router - used as a central bus of communication between modules,
- Inferencing module - used to perform inference on captured images and distinguish weeds from crops,
- Map generation module - designed to create a yield map of the field,
- End user dashboard - used to monitor the mission and robot status.

Apart from these main modules, intermediate communication modules (called Zenoh bridges) are also deployed to enable communication between modules without data loss when the connectivity is lost.

Due to their nature, most modules are tightly coupled with specific tiers of the continuum. The camera driver needs to be deployed on the robot to be able to communicate with cameras on the AgroRob. Zenoh router needs to be in the cloud, so that other modules can use it to communicate with each other. Map generation does not need to happen in real-time, so it's better to perform it in the cloud rather than on limited hardware resources on the edge. Dashboard needs to be accessible by the user and deployed close to the robot, so that when there is a connectivity loss between edge and cloud, the operation of the robot can still be monitored and stopped in case of an emergency.

The most dynamic module, which can roam around the continuum, is the Inferencing module. Depending on the hardware installed on the robot and connection quality in the place of robot's work, it could be deployed either on the edge or in the cloud. The first solution requires a high-performance computer installed on the robot, but works well in rural environments, where connectivity is poor, as all the real-time operations are performed on the edge. On the other hand, using cloud to deploy the inferencing module is a much more scalable solution, which allows the use of a less powerful on-board computer, but requires the connection to be stable and fast.

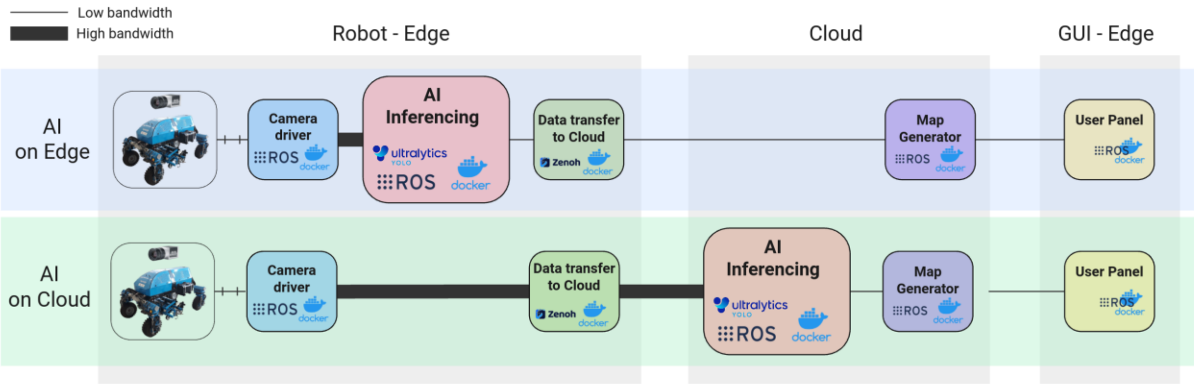

Figure 5 UC1 alternative deployment scenarios

3.1.2 KPIs review

Table 2 UC1 KPIs targets

No.	Project Objective		Target
1	Validation and Mapping of Germinated Plants	*Detection*	90% detection
2	Weeds and Diseases Detection Effectiveness		60% detection
3	Fertilizer Use Reduction	*Efficiency*	400 → 170 l/ha (−57.5%)
4	Herbicide Use Reduction		300 → 60 l/ha (−80%)

3.1.3 KPIs validation

To assess the effectiveness and impact of the autonomous agricultural system, each Key Performance Indicator (KPI) was calculated using real-world field data collected during the sowing, growth, and treatment stages of maize cultivation. The calculations combined sensor-based measurements, geolocation data, image detection models, and agronomic benchmarks to ensure precision and relevance. Detection-related KPIs were validated by comparing system outputs with ground truth observations, while efficiency-related KPIs—such as fertilizer and herbicide usage—were based on area-specific application strategies enabled by the robot's targeted operation capabilities. All calculations reflect practical field conditions and were benchmarked against conventional farming practices to quantify improvements and validate the performance of the robotic system taking advantage of ICOS Meta OS.

KPI 1. Validation and Mapping of Germinated Plants

KPI Target: 90% plant detection accuracy

Achievement Methodology:

The process of validating and mapping the emergence of plants began during the sowing operation. A seed sensor integrated into the precision seeder's sowing section detected and counted the seeds deposited into the soil. Each seed's location was logged with precise GPS coordinates.

After germination, an autonomous vehicle equipped with a specialized camera system performed field passes. The camera captured images of the actual emerged plants, and these were georeferenced to produce a plant emergence map.

By comparing the recorded positions of the sown seeds with the actual positions of the emerged plants, the system was able to validate the emergence rate. The detection algorithm differentiated between emerged and non-emerged positions, enabling a quantitative analysis of the germination process.

Result:
Achieved a plant detection rate of approximately **90%**, confirming the validity of the autonomous plant mapping solution. This matched well with seed manufacturers' germination rate claims and provided valuable insights into sowing quality and seed performance under real-world conditions.

KPI 2. Detection Effectiveness of Weeds and Plant Diseases

KPI Target: 60% detection accuracy

Achievement Methodology:

The detection capability for weeds and diseases was evaluated using the ICOS robot platform equipped with an onboard vision system. The AI model (YOLO-based detector) was trained on field-collected images, manually annotated to distinguish corn plants from weeds. The dataset included over 39,400 crop annotations and nearly 3,700 weed annotations, representing diverse conditions across growth stages and row densities.

Performance was assessed on a held-out validation set. The model achieved precision of 87.4%, recall of 85.7%, and a mean Average Precision (mAP@0.5) of 90.1%. Even under the more stringent mAP@0.5–0.95 metric, the system reached 60.3%, indicating robust performance across detection thresholds. These results confirm that the model reliably identifies both crop plants and weeds in field scenarios, supporting targeted monitoring and precision treatment.

Result:
The robot's AI-driven vision system exceeded the KPI target of 60% detection accuracy. With validated detection rates above 60% under practical conditions, the platform demonstrated competent functionality, sufficient to guide decision-making for precision weeding and plant health monitoring.

KPI 3: Reduction in Liquid Fertilizer Use via Prediction

Target: From 400 to 170 l/ha (**−57.5%**)
Achieved:34–36 l/ha (**91.5% reduction**)

Description:
The ICOS system enables a **precision fertilization approach**, reducing liquid fertilizer consumption through two key innovations:

1. **During sowing:** Fertilizer is applied in localized bands or points near the seeds.
2. **After emergence:** Fertilizer is applied selectively only where plants have emerged, based on actual germination maps.

Calculations:

- **Traditional full-area fertilization:**
  1 ha × 400 l = **400 L/ha**
- **Band fertilization (0.1 m wide band for 75 cm row spacing):**
  Area: 10000 m² / 0.75 m × 0.1 m = 1333 m² = 0.133 ha
  Fertilizer used: 0.133 ha × 400 l = **53.2 L/ha**
- **Point fertilization (90,000 plants/ha, 0.1 m × 0.1 m per plant):**
  Area: 90,000 × 0.01 m² = 900 m² = 0.09 ha
  Fertilizer used: 0.09 ha × 400 l = **36 L/ha**
- **Post-emergence fertilization (85,000 plants/ha):**
  Area: 850 m² = 0.085 ha
  Fertilizer used: 0.085 ha × 400 l = **34 L/ha**

Conclusion:
The use of actual plant emergence maps for fertilizer application has resulted in a reduction from 400 l/ha to **34–36 l/ha**, which constitutes a **91.5% decrease**, greatly exceeding the original KPI goal of a **57.5% reduction**.

KPI 4: Reduction in Herbicide Use via Task Optimization

Target: From 300 to 60 l/ha (**−80%**)
Achieved:18–26.6 l/ha (**91–94% reduction**)

Description:
This KPI refers to reducing the use of plant protection products through **optimized application techniques** enabled by the robot:

- **Mechanical weeding** between rows
- **Selective spraying** between plants using AI-based detection

Calculations:

- Conventional spraying (full-field):
  1 ha × 200 l = **200 L/ha**
- Band spraying (10 cm wide band for 75 cm row spacing):
  Area: 10000 m² / 0.75 m × 0.1 m = 1333 m² = 0.133 ha
  Herbicide used: 0.133 ha × 200 l = **26.6 L/ha**
- Selective spraying (90,000 plants/ha, 0.1 × 0.1 m per plant):
  Area: 900 m² = 0.09 ha
  Herbicide used: 0.09 ha × 200 l = 18 L/ha

Conclusion:
Using robotic selective and band application techniques, the herbicide use has been reduced from 300 l/ha to as low as **18 l/ha**, achieving **91–94% reduction**, again **far exceeding** the targeted KPI of an **80% reduction**.

Table 3 UC1 KPIs review

No.	KPI	Target	Achieved	Result
1	Validation and Mapping of Germinated Plants	90% detection	90% detection	☑ Met
2	Weeds and Diseases Detection Effectiveness	60% detection	60% detection	☑ Met
3	Fertilizer Use Reduction	400 → 170 l/ha (−57.5%)	400 → 34–36 l/ha (−91.5%)	☑ Exceeded
4	Herbicide Use Reduction	300 → 60 l/ha (−80%)	300 → 18–26.6 l/ha (−91–94%)	☑ Exceeded

3.2 UC2 Railway Structural Alert Monitoring system [WSE, FGC]

3.2.1 Functionality validated (DEMO)

Use Case 2 validates the capacity of ICOS to ensure resilient, safety-critical application execution for railway infrastructure monitoring, even under adverse connectivity conditions. Two complementary test cases, presented in the demonstration video, were conducted to validate functionalities with significant impact for the railway vertical.

Test 1: Off-line application execution: Critical Application Execution Edge-Based Cant Monitoring for Rail Safety demonstrated that tilt sensor data (showing cant inclination), processed through an ICOS on boarded edge gateway, continued to trigger safety alerts locally when cloud connectivity was lost. Safety business rule that indicates the alert when the value is exceeded were executed reliably locally (at the edge), ensuring uninterrupted monitoring and alerting of rail cant inclination allowing to exceed the KPI of improving the service availability.

Test 2: Data Integrity: Uninterrupted Data Capture Edge Stores Data & Syncs When Connectivity Restores. This test verified that data from tilt sensor was continuously captured and stored securely on the gateway edge device on boarded to ICOS during a simulated connectivity outage. Once connectivity was restored, all buffered data was automatically synchronized with the cloud, guaranteeing data completeness and integrity.

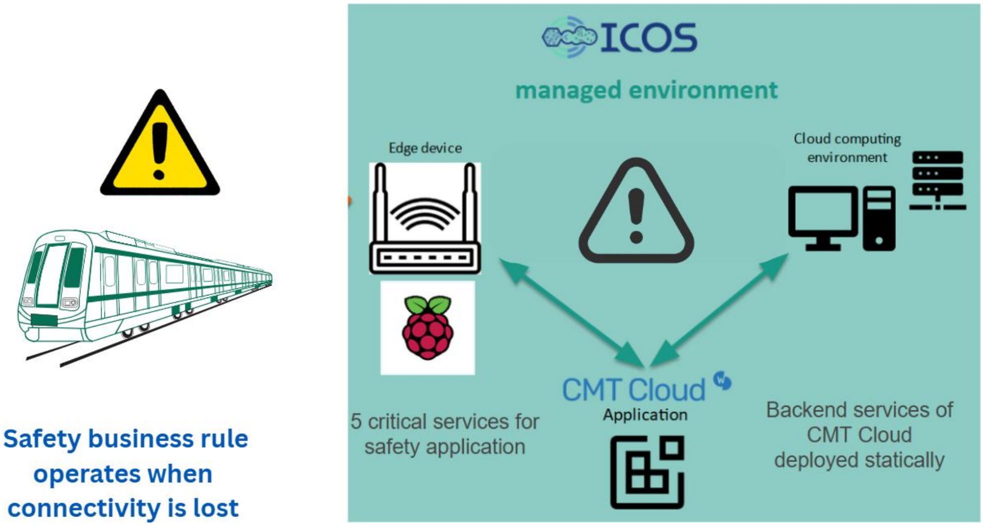

Figure 6 UC2 Functionalities validation

These results confirm that ICOS enables uninterrupted local execution, reliable alerting, and seamless cloud synchronization—key requirements for resilient railway monitoring. Importantly, Use Case 2 not only achieved but exceeded the target KPI of a 10% increase in service availability, delivering 100% availability during testing.

3.2.2 KPIs review

Table 4 UC2 KPIs targets

No.	Project Objectives	Target
1	Reduce Delay	-10% delay reduction
2	Increase Availability	+10% availability
3	Reduce Cybersecurity Threats	-10% cyber security threats

KPI 1: Reduce Delay by 10%
Achieved Result: ~83% Delay Reduction *(from 3s to 0.5s)*

Achievement Methodology:
In the original setup, the delay was primarily caused by the time required for the Edge Device - Cloud Raspberry Gateway (GW) to obtain a threshold breach in the railway cant difference. This process typically took around 3 seconds, including:

- Receiving and forwarding IoT device message in edge device (GW),
- backhaul internet message transport,
- message decoding, data parsing at cloud server,
- executing the safety-related business rule at cloud server, and
- sending the resulting alert to the rail infrastructure operator.

With ICOS, the entire workflow is processed locally **within the edge device**, eliminating the need for cloud communication. As a result, the execution time has been reduced to just **0.5 seconds**.

Impact:
This shift to on-device processing leads to an **83% reduction in delay**, significantly surpassing the initial KPI target of 10%.

<u>KPI 2: Increase Availability by 10%</u>
Achieved Result: 100% Availability

Achievement Methodology:
Availability in this context refers to the system's ability to execute safety-related business rules and trigger the alarm, which depends on the connectivity between the Edge Device (Gateway) and the cloud.

Before ICOS, the execution of these rules required an active internet connection. In the event of a connectivity loss, the service would be disrupted—resulting in **0% availability** during offline periods.

With ICOS, the entire safety logic is processed locally on the edge device, eliminating dependence on cloud connectivity. As a result, the business safety rule remains fully operational at all times, regardless of internet availability.

Impact:
This local execution architecture ensures **100% service availability**, far exceeding the original KPI target of a 10% increase.

<u>KPI 3: Reduce Cybersecurity Threats by 10%</u>
Achieved Result: Significant Risk Reduction Through Edge Execution

Achievement Methodology:
In this context, cybersecurity threats refer to potential attacks where malicious actors could interfere with data reception at the cloud server to trigger alarms (business safety rule) or forge the business rule execution. Any of those scenarios would have a direct impact as rail operations would be stopped. This type of risk is significantly more severe in cloud-based architectures, where compromising the cloud, with public access published and accessible on the internet, can disrupt the entire service.

Before ICOS, critical business logic was executed in the cloud, making it a high-value target. An attack on the cloud environment could affect all connected devices and lead to false alarms, operational disruptions, or data breaches.

With ICOS, the architecture shifts to a **distributed model**, where critical safety rules and functions are executed locally on the edge device. This decentralization **reduces the attack surface** and **isolates potential threats**, meaning that even if the cloud is compromised, the edge devices continue to operate safely and independently.

Impact:

By eliminating the reliance on centralized cloud execution for critical operations, ICOS greatly reduces exposure to cybersecurity threats. This architectural change represents a **significant reduction in risk**, well beyond the targeted 10%. Industry studies show that attacks on cloud infrastructure are 2–3 times more frequent and more disruptive than those targeting edge devices, reinforcing the security advantage of local processing. *

Table 5 UC2 KPIs review

	KPI	Target	Achieved	Result
1	Reduce Delay	10%	83% Delay Reduction	☑ Exceeded
2	Increase Availability	10%	100% service availability	☑ Exceeded
3	Reduce Cybersecurity Threats	10%	Significant Risk Reduction Through Edge Execution	☑ Met

3.3 UC3: In-car Advanced Infotainment and Multimedia Management system [UniTo, CRF]

3.3.1 Functionality validated (DEMO)

IAIMM Application and Business Logic Architecture

The **In-car Advanced Infotainment and Multimedia Management system (IAIMM) UC3** offers innovative media content and tourism-oriented services to enhance the passenger experience while travelling in a vehicle. Among its core functionalities, the IAIMM application relies on **Agentic Generative AI** to enhance the passenger journey. This capability underpins two key functions: the **"Follow Me"** feature, which provides personalized guidance and points of interest recommendations in real time based on user preferences, location, and contextual conditions; and the dynamic generation of travel routes, where the system suggests tailored itineraries by organizing selected points of interest into coherent paths. These routes are built on the basis of previously collected **Points of Interest (POIs)**, selected from reliable sources, and refined according to the explicit preferences expressed by the user. For example, if a passenger indicates a strong interest in cultural sites rather than gastronomic experiences, the system automatically suggests an itinerary prioritising museums, historical landmarks, and heritage sites along the route. In this way, the system not only recommends individual POIs but also organizes them into coherent itineraries tailored to the passenger's profile.

Complementing this, the **"Enjoy Media Together"** service enables collaborative media sharing and remote interactions between passengers and remote interested parties through integrated video calls. This functionality allows users to share experiences and enjoy interactive entertainment while on the move, thereby expanding the scope of in-car infotainment beyond traditional content delivery.

The IAIMM application relies on a microservices-based architecture, designed to operate seamlessly across the cloud–fog–edge continuum enabled by ICOS, as illustrated in Figure 7. At its core, the **Backend service (FastAPI)** implements the business logic, managing communication with the **Frontend (NextJS)**, which in the experiments was deployed on a tablet for simplicity, while in a real deployment it would be integrated into the car's entertainment screen (the so-called radio box). Through this interaction, users access both the RAG-based conversational system and the retrieval of 3D models of points of interest. The conversational pipeline is supported by **LiteLLM**, an open-source microservice that has been integrated into the application's business logic to enable routing across distributed LLM servers. LiteLLM allows the system to transparently contact LLM services regardless of their deployment location—cloud, fog, or edge—and to apply built-in fallback policies when conditions degrade. Specifically, degradation is defined by a timeout of 2 seconds: if this threshold is exceeded, LiteLLM automatically reroutes requests to alternative nodes, ensuring service continuity.

Model selection is performed dynamically based on policies related to network availability and stability, ensuring that inference requests are always directed to the most suitable resource. In this architecture, **high-capacity models** (e.g., Qwen3 32B) are deployed in the **cloud**, **mid-size models** (e.g., Qwen3 13B) in **fog edges**, and **lightweight models** (e.g., Llama 3.2 3B Instruct quantized) directly in the vehicle's Car PC.

The transition between small, medium, and large LLMs remains completely transparent to the user. The system automatically selects the most appropriate node and adapts the task to the capabilities of the available model—for example, simplifying prompts when executed by a lightweight model at the edge, while relying on cloud resources for more complex reasoning. In this way, the quality of interaction is preserved independently of the underlying infrastructure. Complementary services further strengthen the architecture: the Conversation Database can be relocated across nodes according to security policies, the MinIO Bucket in the cloud stores the complete catalogue of 3D models, and Redis caches at fog nodes provide rapid access to up to five nearby POIs, ensuring responsiveness even in low-connectivity conditions.

The integration with ICOS enables the dynamic relocation and orchestration of these services, ensuring continuity and resilience in real-world conditions. By default, LiteLLM routes requests to cloud-hosted models, maximizing performance under stable connectivity. In case of network degradation, requests are redirected to fog nodes, and ultimately to the Car PC if necessary, ensuring uninterrupted operation while preserving **Quality of Service (QoS)** and **Quality of Experience (QoE)**. For 3D content, caching mechanisms on fog nodes guarantee availability of a minimum set of POIs even without cloud access, while central cloud storage ensures scalability. Security policies orchestrated by ICOS further guarantee that sensitive components, such as the Conversation Database, are relocated to secure nodes whenever threats are detected. Altogether, this architecture illustrates how the IAIMM application integrates heterogeneous services under ICOS, validating the key functionalities required by the use case and demonstrating the practical value of features such as "Follow Me" and "Enjoy Media Together".

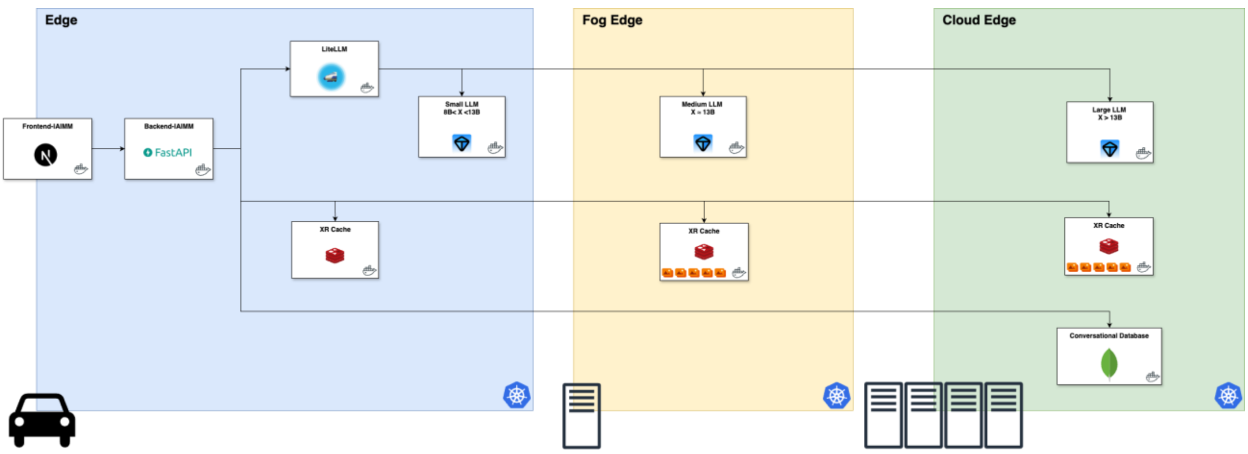

Figure 7 UC3 Business logic architecture of the IAIMM application

Video Demonstration

The demonstration illustrates the deployment of the IAIMM application integrated with ICOS in a real vehicle environment (a Jeep Renegade) [4]. Although the use case is conceptually framed within the context of autonomous driving, the demonstration was carried out using a person-driven vehicle, due to the difficulty in using a self-driving car in the city traffic at the time of testing. The first part of the video presents the vehicle setup, the application interface, and an overview of its functionalities developed by UniTo. This is followed by a detailed view of the vehicle environment and the configurations prepared by Stellantis-CRF. The demonstration then shows the application in operation, highlighting its main functionalities, and concludes with an explanation of the infrastructural setup and its integration within the ICOS ecosystem developed by UniTo, as illustrated in Figure 8.

Figure 8 UC3 IAIMM application running on the in-vehicle Car PC (Jeep Renegade)

The key added values provided by ICOS in this context are offline execution and the ability to define and enforce custom policies. By leveraging the ICOS ecosystem, it becomes feasible to deliver a high-quality, personalized tourist entertainment service with reduced operational complexity. The video also demonstrates how critical services, such as LLM models of different sizes (3, 12, 30, and 80 billion parameters), are seamlessly migrated in real time to the most suitable nodes identified by the ICOS **MatchMaker**, according to custom service-defined metrics.

Once these decisions are taken, the **Job Manager** enforces the directives of the MatchMaker, ensuring that service re-deployments and migrations are correctly executed across the continuum. Together, these ICOS components enable a fully automated and dynamic system that adapts to run-time conditions defined during the application deployment phase. It's possible to see the distribution of LLM services in the network of nodes in the Figure 7.

The activities related to node configuration, application development, and testing were carried out by the University of Turin (UniTo) in close collaboration with Stellantis-CRF, ensuring that both the research and industrial perspectives were fully integrated into the demonstration. It is possible to consult the video produced at Section 7.

Offline Application Execution and Custom Policies

All the requirements defined for UC3 — including interoperability across heterogeneous devices, operating system compatibility, minimum hardware specifications, container orchestration, networking, security, and latency management — have been successfully satisfied, thereby confirming the effectiveness of the IAIMM use case. A summary of these requirements is reported in Table 6, which served as the baseline for the validation activities. On this basis, the case study can be considered a success, as the IAIMM application demonstrated its ability to meet the technical and operational conditions necessary for deployment within the ICOS ecosystem.

One of the most relevant achievements of UC3 is the ability to execute applications and core services under offline or degraded connectivity conditions, thereby ensuring service continuity and preserving both QoS and QoE. Information and computation are distributed across three layers: cloud nodes, which host the complete catalog of 3D models and more complex datasets; fog edge nodes, which cache a limited subset of resources to ensure low-latency access; and in-vehicle edge nodes. By default, the IAIMM application connects to the cloud, which guarantees full access to all available content. When cloud connectivity is not available, fog edge nodes provide access to at least five nearby Points of Interest (POIs) and their associated 3D models, ensuring continuity of service. Finally, the in-vehicle Car PC offers local fallback functionalities, maintaining a minimum level of responsiveness even in fully offline conditions. The ClusterLink service enables robust connectivity between nodes, ensuring seamless synchronization of services and data across cloud, edge, and device layers. This distributed balance minimizes service interruptions, maintains low-latency interactions, and sustains consistently high QoE even in intermittent connectivity conditions. In addition, the IAIMM application allows real-time monitoring of service quality, informing passengers whether their experience is powered by optimal infrastructure components or fallback local services, thereby increasing transparency and awareness.

Equally important is the capability to define and enforce custom policies, which dynamically regulate the execution of critical services and directly support QoS and QoE preservation. This is particularly relevant in the Enjoy Media Together feature, where responsiveness and synchronization are essential for collaborative media experiences. Policies were expressed using advanced service-level metrics, such as the p95 latency per token provided by Hugging Face TGI (a metric that will be described in detail in the following Section 3.3.2 - KPIs review). This metric captures both computational load (queuing, batching, processing) and network-induced delays and was selected as the primary indicator of user-perceived responsiveness in interactive applications. Thresholds defined on this metric allowed the ICOS MatchMaker component to automatically migrate critical services — such as LLM servers — from overloaded or distant nodes to better-performing ones. In this way, QoS parameters (e.g., latency, throughput, availability) remained within acceptable ranges, while QoE was safeguarded through smooth and responsive interactions. At the same time, the IAIMM application provided user-facing visibility into service quality, notifying passengers whenever the system transitioned between higher-performing cloud/edge resources and constrained local services. While ICOS ensured automated orchestration in the background, the application communicated these adaptations transparently to end users.

Table 6 UC3 requirements as per M36

ICOS Requirements	Use Case Status	Comment
Interoperability: The computational infrastructure and devices must be capable of seamless integration. This necessitates standard protocols, a shared network infrastructure, and interfaces.	Satisfied	Data streaming from the IoT to near edge to the Cloud environment has been successfully proven in UC3.
ICOS is engineered to allow its control plane to operate on any OS that officially supports Docker or Kubernetes technologies. However, CentOS and Ubuntu are the recommended operating systems.	Satisfied	The Edge devices included in the automotive Use Case use Linux Ubuntu and MacOS with Linux virtualization for Docker.

ICOS Requirements	Use Case Status	Comment
The minimum hardware requirements for the computational nodes are 512MB of RAM, 2GB of free disk space, and support for CPU architectures such as aarch64, x86_64, and amd64.	Satisfied	Minimum hardware requirements for the computational node are met.
ICOS manages itself using a Container Orchestration approach, requiring Docker Engine (version 18 or higher) or one of the Kubernetes distributions (k3s, k0s, k8s), along with Helm.	Satisfied	Docker distributions integrated: Docker engine and Podman engine. Kubernetes distributions integrated: Kubernetes light version (k3s), Kubernetes light version wrapper to run k3s called k3d, kind, rke2.
In IT-1, all computational nodes need to be visible under the same network and have a series of ports opened.	Satisfied	Hardware of UC3 is connected under the network provided by modem. Connection with Athens testbed is established via OpenVPN.
Security: The integration of various systems and devices raises significant security concerns. In addition to the security layer provided by ICOS, additional measures should be implemented to protect data integrity and privacy at the hardware level.	Satisfied	UC3 involves the use of personal data and requires medium security levels. The UC integrates ICSO security policy to protect database conversations.
Latency: For edge computing and IoT devices, low latency is essential for real-time data processing and decision-making.	Satisfied	Latency is very important for QoS and QoE. To reduce the latency to minimum, 5G network will be used, as well as anticipating areas of slow network connectivity to optimise data offloading when the connection is stable. The IAIMM app incorporates improvements based on the micro-service position in the ICOS nodes, the implementation of a specific proxy for LLM requests, and 3D model caching to reduce latency.

3.3.2 KPIs review

Table 7 UC3 KPIs targets

No.	Project Objectives	Target
1	Reduce Delay	-10%
2	Service Availability	+20%
3	Security and Privacy	-10%

The evaluation of the IAIMM use case has been conducted through a set of Key Performance Indicators (see Table 7) designed to demonstrate how the ICOS ecosystem enhances service delivery across the cloud–edge continuum. These KPIs assess improvements in responsiveness, service availability, and security, while ensuring high levels of Quality of Service (QoS) and Quality of Experience (QoE).

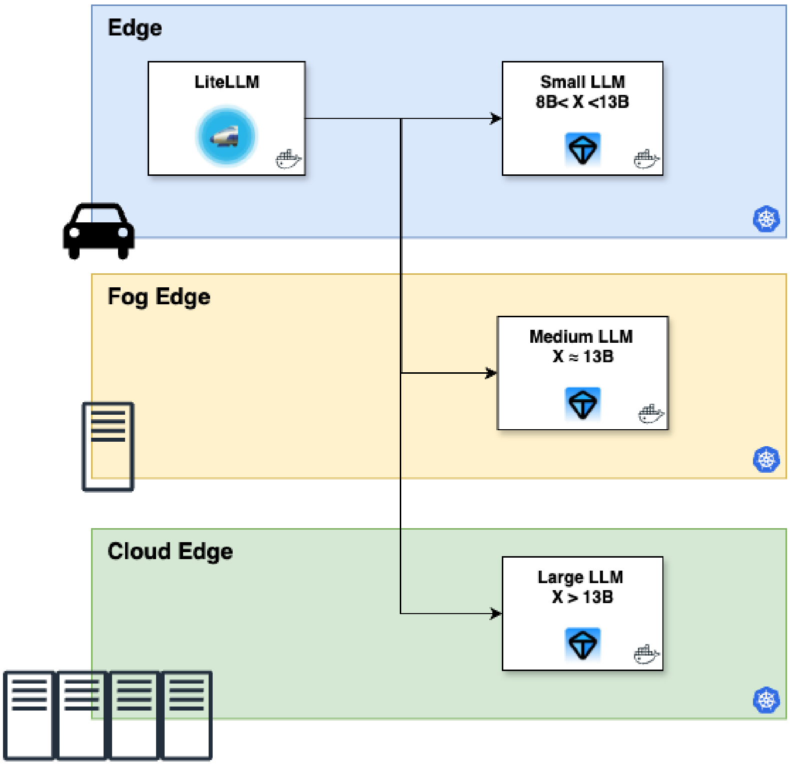

Figure 9 UC3 Cloud–fog–edge app architecture for LLM server services

KPI 1. Delay Reduction

KPI Target: -10% delay reduction

Achievement Methodology:

System responsiveness was primarily evaluated on the basis of Large Language Model (LLM) services, identified as the most latency-sensitive component of the infotainment application. These services were deployed as self-hosted micro-services across IoT nodes within the ICOS continuum, spanning from cloud-based VPS to fog edge nodes in urban areas and in-vehicle Car PCs, as illustrated in Figure 9. In Figure 10 (see below) an example is provided of an LLM server being redeployed from one node to another after the violation of the defined policy conditions.

The key metric used for policy enforcement was the **p95 Latency per Token** (**"tgi_request_mean_time_per_token_duration"**), which measures inter-token streaming latency normalized by output length. This metric was selected as the most representative of user-perceived responsiveness in conversational and streaming contexts, as it captures both system load conditions (queueing, batching, compute) and network effects. During the experiments, custom ICOS policies relied exclusively on this per-token metric to automatically migrate LLM services across the most suitable IoT nodes. Service continuity during migration (during the ICOS remediation action called **"redeploy"**) was preserved through the LiteLLM proxy, which temporarily rerouted requests to a local fallback endpoint. This mechanism masked the ≈2-second timeout typically introduced during relocation, ensuring uninterrupted operation and preserving QoE for passengers. Latency in this case was mitigated through a caching mechanism: each fog edge node locally stored at least five POIs with their associated 3D models, which were delivered directly to vehicles entering the node's coverage area. This guaranteed a minimum level of continuity in low-connectivity conditions, while the majority of the XR assets remained available from cloud or higher-capacity nodes.

Results:

The reduction of delay was computed following the standard methodology for QoS improvement assessment defined in ITU-T Rec. E.800 and ETSI GS NFV-TST 001[5][6], using the formula:

$$\text{Delay Reduction } (\%) = \frac{L_{cloud} - L_{ICOS}}{L_{cloud}} \times 100$$

- In the centralized cloud-only architecture, the p95 per-token latency was observed at approximately **0.55 s/token (latency)**.

- With the ICOS-enabled distributed architecture, the same metric decreased to around **0.33 s/token (latency)**.

Applying the formula:

$$\frac{0.55 - 0.33}{0.55} \times 100 \approx 40\%$$

This corresponds to a delay reduction of **~40%**, significantly surpassing the original target of 10%, reported in Table 7. These findings are consistent with empirical studies on fog computing, which report average response time improvements of 20–30% compared to cloud-only deployments [6][7]. They also align with Hugging Face's benchmarking guidelines, which emphasize p95 per-token latency as the most representative metric of user-perceived responsiveness in streaming and conversational applications [9][10].

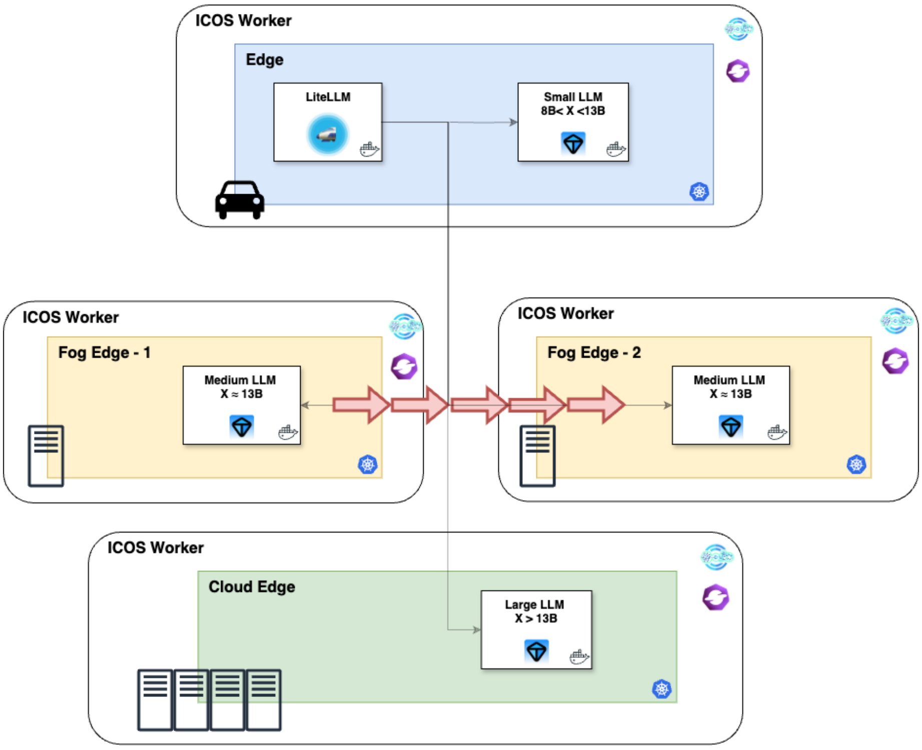

Figure 10 UC3 Dynamic relocation of LLM services across IoT nodes

KPI 2. Service Availability

KPI Target: +20% of potential service available with the same use of resource

Achievement Methodology:

This KPI evaluated the system's ability to maintain critical service availability with identical resource usage, comparing a centralized cloud architecture with a deployment managed through ICOS.

In a standard centralized model, service availability is tightly coupled with stable connectivity and centralized resource reliability. A distributed alternative without ICOS would imply:

- high operational overhead for maintaining edge nodes,
- manual deployment/migration of services,
- lack of real-time orchestration intelligence.

ICOS, instead, enables:

- automated service orchestration based on declarative policy definitions and dynamic node assessments,
- localized execution of selected services, such as subsets of 3D point-of-interest models cached on fog edges (up to 5 POIs per edge), ensuring continuity even in low-connectivity scenarios (see Figure 10),
- policy-driven migration of critical services such as LLMs across cloud, edge, or local nodes depending on real-time QoS metrics (e.g., latency, throughput) and their impact on end-user QoE.

In addition, the IAIMM application itself provides end-user visibility into service quality, informing passengers whether their current experience is powered by optimal cloud/edge services or by fallback local services in degraded conditions.

Results:

Under the evaluated test conditions, the integration of ICOS with the IAIMM application ensured **100% availability** of LLM-based services, as illustrated in Table 8. This was achieved through policy-driven migration across cloud, fog, and edge nodes, which allowed uninterrupted operation even during network disruptions. For XR services based on 3D models of points of interest, availability was supported by a caching mechanism deployed on distributed fog nodes. While this approach proved effective within the tested environment, large-scale deployment of such services—requiring both the production of a significant number of 3D models and the resources for their delivery—remains an open challenge. Consequently, experimentation has so far been limited to a representative urban scenario, with the expectation that broader trials will confirm the scalability of the approach.

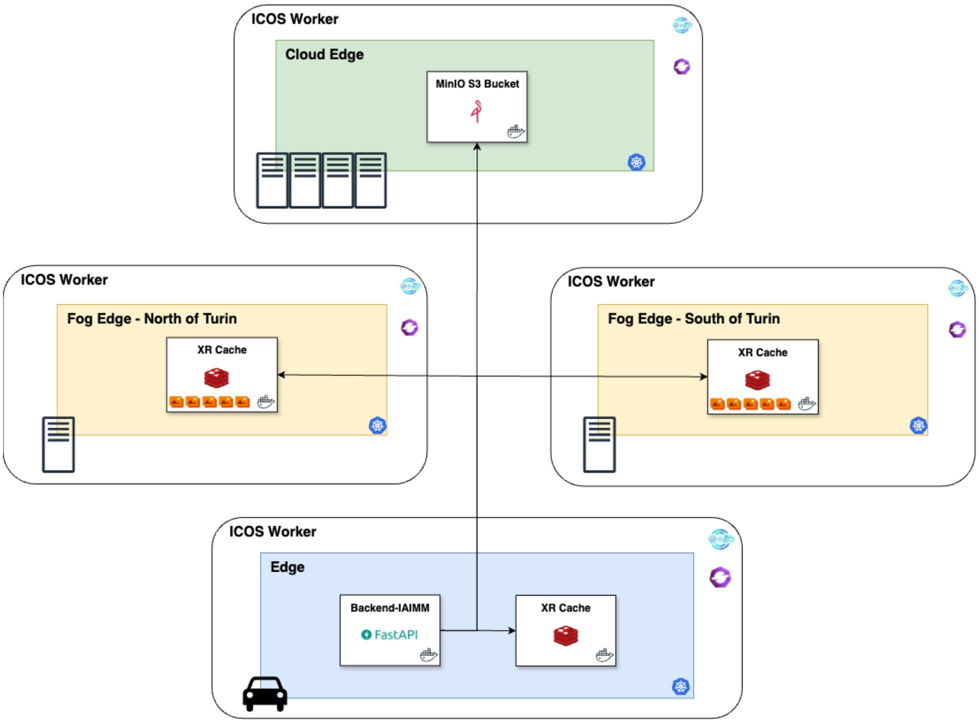

Figure 11 UC3 Caching of 3D models across the cloud–fog–edge continuum integrated within ICOS.

KPI 3. Security and Privacy

KPI Target: -10% cybersecurity threats in cloud

Achievement Methodology:

The infotainment application deployed in the vehicle was secured through authenticated access to the conversational interface, with all data stored locally and only periodically synchronized with the cloud to minimize exposure.

Beyond this baseline protection, ICOS policies were applied directly to the database service used to store conversations. These policies continuously monitored the security posture of the nodes hosting the database, using the **Wazuh Security Configuration Assessment (SCA)** framework. The SCA module evaluated the system configuration against predefined security controls and produced a qualitative score, classifying nodes into three levels of compliance: high, medium, or low.

Whenever the security level of a node hosting the database dropped from high to medium or low, ICOS triggered the relocation of the service to another node with a stronger compliance profile. This ensured that critical data was always hosted in an environment aligned with security best practices, without disrupting service continuity for the end user.

During the evaluation, a node initially classified as high was flagged as medium by Wazuh due to misconfigurations detected in the security baseline. In response, ICOS automatically migrated the database service to another node classified as high, ensuring both data integrity and service availability.

Result:

Through this adaptive security mechanism, the number of high-risk situations was reduced in comparison to a centralized set-up, surpassing the initial 10% reduction target, as illustrated in Table 8. The combination of continuous monitoring with Wazuh and automated policy enforcement by ICOS proved effective in strengthening both security and privacy guarantees within the application.

Impact

The integration of ICOS into the IAIMM use case has demonstrated measurable improvements in latency, service availability, and security for AI-powered infotainment and entertainment systems in mobility scenarios. With respect to latency, the adoption of policy-based automation enabled critical LLM services to be dynamically migrated across IoT nodes of different capacities (from cloud VPS to urban fog edges down to in-vehicle Car PCs). This was achieved using the per-token latency metric as the primary driver for relocation, ensuring a fluid streaming experience and meeting the KPI target for delay reduction.

Regarding service availability, the combined use of ICOS orchestration and the IAIMM application allowed uninterrupted service continuity, even in cases of connectivity loss during vehicle operation. Under test conditions, LLM services achieved almost 100% continuity thanks to the integration of LiteLLM as a local proxy during migration events. For XR services, continuity was ensured through caching strategies, whereby fog nodes locally stored a subset of POIs, thus guaranteeing a minimum level of service in degraded conditions.

Finally, in the domain of security, ICOS enabled proactive enforcement of policies on critical services, isolating and relocating them to more secure nodes whenever potential risks were detected. This mechanism, supported by the integration of Wazuh for monitoring, reduced the exposure of sensitive components and exceeded the KPI target for threat mitigation. Overall, ICOS has proven to be a key enabler for resilient, adaptive, and secure service delivery across the cloud–edge–IoT continuum, effectively improving both Quality of Service (QoS) and Quality of Experience (QoE) in the IAIMM use case.

Table 8 UC3 KPIs review

No.	KPI	Target	Achieved	Result
1	Reduce Delay	-10%	~40%*	☑ Exceeded
2	Service Availability	+20%	~100%*	☑ Exceeded
3	Security and Privacy	-10%	Significant Risk reduction	☑ Met

*Under test condition. The number is referred in particular for the LLM server, considerate as critical service.

3.4 UC4: Energy Management and Decision Support system [SSEA]

UC4 focuses on optimising domestic energy use, with Heat Pumps, PV systems, and storage playing key roles in achieving net-zero emissions by using Advanced Machine Learning techniques for energy forecasting.

Challenges: Ensuring data protection and security and providing customized energy solutions

Expected Benefits: By harnessing Cloud and Edge capabilities for real-time solutions, ICOS aims to reduce data transfers and latency, increase security, and improve resource flexibility tailored to each customer's specific needs, ultimately enhancing customer satisfaction and retention.

3.4.1 Functionality validated (DEMO)

As UC4 involves the use of personal data at a highly granular level, data security, data privacy, and data integrity are of fundamental importance UC4 validates the capacity of ICOS to increase security and privacy of user data. The video demonstration involves one node in which the ICOS application is deployed, and a second node which acts as a ‘backup’, running on an independent network provided by a router and SIM card. The demonstration establishes the protocol in which a simulated security event occurs, lowering the security score of the main node. The established security policy within the app descriptor then migrates the application to the second ‘backup’ node and simultaneously terminates operation on the ‘compromised’ node.

For each ICOS node, a Security Scan (Wazuh) Agent is installed that is scheduled to periodically complete an SCA scan of the nodes and reports the SCA score to Grafana. The node starts with a default score (a set threshold). If there is any action to harden the note (increase score) or compromise the node (lower score), the SCA scan detects this and reports to the user through Grafana. The security policy in the Policy Manager works in such a way that if the node is under risk of being compromised (either by external or internal malicious actor or unintentionally), the SCA scan would show the score below the default (below the threshold) and that would send the notification to the Policy Manager to trigger the redeployment loop, that is redeploying the application workload to an uncompromised (secure) node.

The security demo in UC4 demonstrates the following key functionalities:

- **Data Integrity and Management:** Ensure that no data loss occurs during a security event, data collection will be terminated in the compromised node, and the application will be redeployed to the backup node.
- **Limiting Disruption to the homeowner:** Ensure that the security policy operation will not affect the homeowners use of the EMDS.
- **Reliable Alerting:** Detection and mitigation of any threats as regulated by security scores.
- **Ensuring Data Security and Privacy:** Verify that a high level of data security and privacy is maintained at all stages with minimal data transfer and data exposure That would otherwise be incurred via cloud transfer.

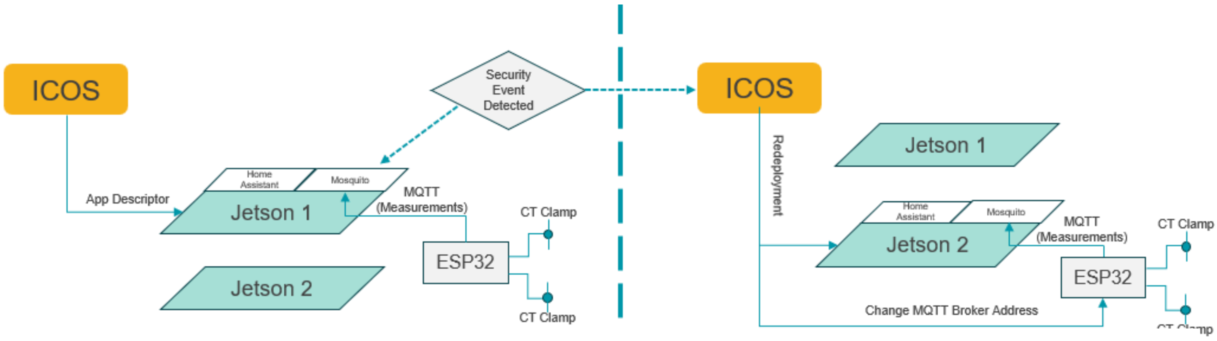

Figure 12 UC4 Security Demo Flow Diagram

3.4.2 KPI's review

Table 9 UC4 KPIs targets

Project Objectives	Detailed Use Case Description	Requirement for ICOS	Target
Security and Privacy	Ensure a high level of data security and data privacy is maintained at all data collection, processing, and storage stages including data synchronization and integrity.	ICOS security and data management layer feature implementation should ensure the detection and mitigation of malicious activity and optimal data management and storage.	-10%
Flexibility Increase in potential service available with same use of resources	Improve the provision of trend identification and optimised predictions of energy to flatten the demand curve by removing demand at peak time and boosting energy usage at nighttime	ICOS should provide mechanisms to set up automated decisions tailored to customer needs by implementing trustworthy AI models including reinforcement learning and federated learning at the EDGE level.	+20%
Delay Reduction in time to configure and manage cloud infrastructure Latency reduction and real-time prediction	Real-time demand-supply predictions are vital to understand electricity usage and consumption to ensure reliable and interrupted services.	ICOS Reinforcement Learning capabilities should provide an increase in configuration and deployment times.	-30%

KPI 1. Security and Privacy: 10% Decrease in cyber security threat

Achievement Methodology:

Within this Use Case, cybersecurity threats reference potential attacks where a malicious actor could attempt to gain access to the data retained in the edge components deployed in the test residences. UC4 involves recording of real time consumption data at a 60 second interval given the granularity of the data, it is therefore possible to understand the occupancy level of the house, risking personal security e.g. risk of break-ins in an empty home.

Before ICOS, this process was carried out in the cloud, with information taken from the various smart devices in the homes, sent via API to the cloud, making the cloud database a high value target. In this scenario, live monitoring of the data or the devices are not included.

With ICOS, edge devices are continuously assessed through a Security Configuration Assessment (SCA) framework, which benchmarks each node against established security controls and assigns a classification of high, medium, or low risk. This decentralised architecture eliminates the cloud as a single point of failure, enhancing overall system resilience by ensuring that data is retained locally on edge nodes. Additionally, ICOS provides a mechanism for termination and redeployment in response to unauthorised access or other security anomalies, further strengthening the integrity of the operational environment.

Impact:

ICOS introduces a decentralised approach to handling EMDS operations, significantly lowering vulnerability to cyber threats by avoiding dependence on cloud infrastructure, which is statistically more prone to frequent and severe breaches. It also strengthens security with the use of the Wazuh Security Configuration Assessment (SCA) framework and actively relocates compromised edge nodes to safer environments when risks arise, using real-time monitoring. This strategy not only surpassed threat reduction targets but also enhanced the reliability and responsiveness of services across cloud, edge, and IoT layers.

KPI 2. Flexibility: 20% increase in potential service available with same use of resources

Achievement Methodology:

This KPI evaluates how UC4 can improve potential service available with same use of resources, this is delivered using Machine Learning. These mechanisms have been provided both at the controller level, reachable from the ICOS shell, and at the edge level as 'AI Support Containers'. UC4 deploys the application in an app descriptor and the AI in controllers where AI support runs as a service. Federated Learning is also provided at the edge level but is restricted to MetaOS metrics in its current version.

Before ICOS, the Average cloud latency ranges from 100ms to 1,000ms, depending on network congestion and server location. As a baseline check, data taken from customer residences in Ireland via cloud-based EDMS is called via API to a central cloud location located in Australia in a process that takes **~300ms** (WonderNetwork). We have assumed the lower end of 100ms in the KPI calculation below.

With ICOS, the entire process is executed directly on the edge device, removing any dependency on cloud-based communication. This localised handling has resulted in a rapid execution time with the overall time taken to collect the data from the smart devices e.g. smart meter, inverter, PV, Battery etc, to the edge device is ~**0.5-5ms**.

$$Increase\ in\ potential\ service\ available\ (\%) = \frac{L_{cloud} - L_{ICOS}}{L_{cloud}} \times 100$$

$$Increase\ in\ potential\ service\ available\ (\%) = \frac{100 - 5}{100} \times 100 = 95\%$$

Impact:

Decreasing communication and computation delay is key to unlocking wider aggregation of residential scale flexibility with some flexibility services such as frequency response requiring <10ms response times. By moving EDMS to the edge both service available is increased and ability to engage in grid stability and power price flexibility is increased.

KPI 3. Delay: 30% Decrease in Configuration Time

Achievement Methodology:

This KPI evaluates how with the use of ICOS, UC4 can decrease configuration time. Given the EMDS is a time sensitive operation, a faster onboarding time benefits both the EMDS provider enabling them further capacity to provide system services, and the customer enabling them to access their energy recommendations earlier, resulting in faster return on investment.

Before ICOS, for the EMDS to provide recommendations on energy consumption, a learning period must take place where electricity consumption data is analysed by a machine learning model for training. This period requires up to **447 hours** of minute-by-minute granularity data.

With ICOS, the configuration of future households is assisted by previous households and significantly reduces the overall time for configuration and operation of the EMDS past the 20% target. The minimum data needed for the Reinforcement Learning case with additional fine tuning of the Transfer Learning, would be **~40 hours** of minute-by-minute granularity data.

$$Decrease\ in\ Configuration\ Time\ (\%) = \frac{L_{cloud} - L_{ICOS}}{L_{cloud}} \times 100$$

$$Decrease\ in\ Configuration\ Time\ (\%) = \frac{447 - 40}{447} \times 100 = 91\%$$

Impact:

The implementation of ICOS within UC4 has significantly improved the efficiency of the EMDS configuration process. Previously, onboarding new households required a learning period during which electricity consumption data was analysed by a machine learning model—this could take up to 10 days before energy recommendations could be generated. With ICOS, however, the system leverages data from previously configured households to assist in setting up new ones. This dramatically reduces configuration time, which is particularly beneficial given the time-sensitive nature of EMDS operations. As a result, EMDS providers can increase their capacity to deliver system services, while customers gain quicker access to their personalised energy recommendations.

Table 10 UC4 KPIs review

No.	KPI	Target	Achieved	Result
1	Security and Privacy	-10%	100% detection and mitigation of threats and anomalies in security demonstration	☑ Exceeded
2	Flexibility Increase	+20%	+95% increase in service available using the same resources.	☑ Exceeded
3	Delay Reduction	-30%	-91% Decrease in configuration time	☑ Exceeded

4 Use Cases Feedback

This section presents feedback gathered from all ICOS use cases — including four internal use cases and 5+15 Open Call projects — at the end of the validation process. The feedback is organized into two parts: negative feedback, highlighting challenges and limitations encountered, and positive feedback, showcasing strengths and successful aspects of the platform.

As described in Section 5, many of the improvements made to ICOS during the project were driven directly by this feedback. These targeted enhancements have not only resolved critical issues but also strengthened the platform's usability, stability, and scalability. The outcome is a strong perspective for future exploitation, with ICOS now better positioned for adoption in industrial, research, and commercial environments.

4.1 Positive feedback

- Platform robustness — ICOS was considered functional and stable once configured correctly, even in complex deployments.
- Responsive technical support — Quick assistance via Mattermost and other channels, with many issues resolved rapidly during the project.
- Timely bug fixes and updates — Problems were addressed in subsequent releases, often with added functionality (e.g., ARM support, improved CLI commands).
- Comprehensive monitoring tools — Integration with Thanos, Telemetruum, and Grafana enabled effective application monitoring.
- Secure, simplified external service access — ClusterLink allowed connectivity without requiring unified Kubernetes networks, maintaining security.
- Flexible deployment models — Ability to define hierarchical infrastructures (cloud, fog, edge) adapted to specific application needs.
- Automated resilience mechanisms — Custom policy definitions enabled automatic relocation of critical services in case of failures or attacks.
- Support for constrained devices — Minimal container deployments possible on limited-resource hardware.
- Functional workarounds for non-native device support — For example, MQTT-based solutions for mobile/wearable endpoints worked effectively.
- Improved onboarding process — Documentation, onboarding scripts, and CLI enhancements (persistent authentication, lifecycle commands) made setup easier over time.
- Feature enhancements during the project — Added lifecycle commands, persistent authentication, namespace flexibility, and ARM-compatible builds.
- Cross-domain potential — Recognized applicability across manufacturing, health, mobility, public safety, and other sectors.

4.2 Negative feedback

1. Deployment & Configuration Complexity
   - Steep learning curve for onboarding new users and deploying ICOS components (Core, Controller, Agents, Workers, Keycloak, etc.), especially outside the ICOS testbed.
   - Manual and error-prone configuration of ClusterLink peers, namespaces, labels, node roles, and container affinity rules.
   - Lack of automated workflows for deployment and onboarding (CLI and GUI support limited).
   - Need for template-based blueprints and ready-to-use examples for common deployments.
2. Documentation Gaps
   - Incomplete or outdated documentation for Dynamic Policy Manager (DPM), CLI onboarding, advanced configuration (e.g., ingress, external service exposure), and integration workflows.
   - Missing clear troubleshooting guides for common errors.
   - Lack of interactive tutorials, videos, or role-specific guides.
   - Poor feedback from CLI when manifests are invalid or malformed.
3. Namespace & Resource Management Issues
   - Complexity in managing multiple namespaces for microservices; namespace overrides exist but are cumbersome.
   - Persistent Volumes cannot be deployed directly in app descriptors; require external management.
   - No native discovery service for intra-namespace communication.
4. Observability & Troubleshooting Limitations
   - Limited logs for failed deployments; no persistent storage for error logs.
   - Error messages often not descriptive enough to pinpoint issues.
   - Lack of a single diagnostic interface to validate ICOS component status.
   - Difficulty tracing root causes for orchestration and communication issues.
5. Infrastructure & Network Issues
   - DNS resolution failures, especially with certain routers/ISPs flagging ICOS domains as suspicious.
   - Problems with ClusterLink setup and service communication.
   - Resource constraints on small/edge devices (e.g., Raspberry Pi disk I/O bottlenecks).
   - No native orchestration support for mobile/wearable devices.
6. Stability & High-Load Performance
   - Application redeployment failures when nodes go offline.
   - Migration mechanisms blocked if failing node cannot confirm.
   - Lack of large-scale stress testing tools and real-world load simulations.
   - Scalability under high concurrency and data volume remains unvalidated.
7. Security Concerns
   - Storing secrets directly in manifests without secure vault integration.
   - Some services not exposed by default, requiring manual manifest changes.
   - Gaps in authentication and policy configuration documentation.

5 ICOS Platform: Final Release Overview

5.1 Final Architecture and Components

As highlighted in Table 9 of Deliverable D5.3 [12], the ICOS final release (M31 – March 2025) delivered most of the functionalities expected by the ICOS plan.

The architectural foundation established in previous deliverables — D2.2 [13] and D2.4 [14] — remains valid, defining two primary actors. The ICOS Controller is responsible for managing the continuum (tracking the current system topology and availability) and overseeing the runtime (deploying and monitoring application execution on demand). The ICOS Agent receives instructions from the ICOS Controller and translates them into infrastructure-specific tasks to execute offloaded user applications, handling code execution, data access, and telemetry collection.

The ICOS Controller is the more complex of the two actors and has been designed using a three-layer architecture:

- The Meta-Kernel Layer manages the continuum, the runtime, and collects telemetry.
- The Intelligence Layer supports AI model training, testing, deployment, maintenance, and updates across the edge/cloud spectrum.
- The Security Layer ensures the security of ICOS users, resources, and applications at all times.

To support the data needs of these three layers, the ICOS architecture has been enriched with an additional transversal Data Management Layer.

Although the architecture originally defines a set of components for each layer, slight modifications were made during implementation to overcome development complexity without altering functionality. The final version of the Meta-Kernel Layer components is described in Section 4 of D3.3 [15], while the components of the other layers are detailed in the respective Sections of D4.2 [16].

During the final months of the project, no significant changes were made to the architecture or its components. Use Case partners and Open Call partners validated the ICOS Final release providing valuable feedback used to further improve the software (detailed in Section 5.2).

5.2 Enhancements driven by Validation.

The continuous testing and validation of the ICOS software carried on by the Use Cases allowed to improve the software by fixing issues and implementing suggested features. This process was enabled and made effective and efficient by the CI/CD process set-up in WP5 that ensured reduced times between the implementation of the code and the deployment of that code in the ICOS testbeds.

This feedback-based improvement process was more active and evident in the last part of the project, after the ICOS Complete release (M31 - March 2025, D5.3 [12]). In fact, several improvements to the ICOS codebase have been done as result of the feedback received from Use Cases that validated the final release of ICOS. The most relevant enhancements delivered after M31 are:

- support for static Kubernetes namespaces.
- increased maximum App Descriptor size.
- report ClusterLink and Zenoh statuses in Telemetry DB.
- fixes and usability improvements to the Shell GUI and Policy Manager GUI.
- avoid infinite redeployment attempts.
- added new policies based on device and peripherals availability, and resources (CPU, memory) usage.
- fix parsing of policies with invalid fields.
- add support for deploying in Docker Swarm nodes in Nuvla;

- fixes credentials cache expiration in Nuvla Deployment Manager;
- added automated Zenoh deployment;
- support for running privileged containers;
- improved predictions of metrics;
- improved telemetry dashboards;
- added Continuum View and Auditing telemetry dashboards.

Overall, this process demonstrated to be very effective in improving the ICOS System and make it more robust and usable by real users.

6 Conclusions

This deliverable concludes the work carried out in Work Package 6, presenting the final results, lessons learned, and outlook for ICOS. Over the course of the project, all defined KPIs were successfully validated, confirming the maturity and effectiveness of the ICOS platform. The work has demonstrated the platform's ability to address key challenges related to scalability, integration, and usability. At the same time, it has also highlighted areas that remain open for further refinement, particularly in strengthening interoperability with external systems and exploring additional automation features.

Overall, the project has achieved its objectives and proven the business potential of ICOS as a robust, future-oriented solution. Feedback from Use Cases has been instrumental in shaping improvements, ensuring that the platform is aligned with real-world needs and practical adoption scenarios.

Looking ahead, the next steps involve capitalizing on the platform's strengths to drive future exploitation and market uptake. This includes extending ICOS capabilities beyond the current scope, exploring partnerships for commercialization, and continuing technical enhancements in line with the project roadmap. With these directions, ICOS is well-positioned to generate long-term impact, both technologically and in terms of business value, beyond the lifetime of the project.

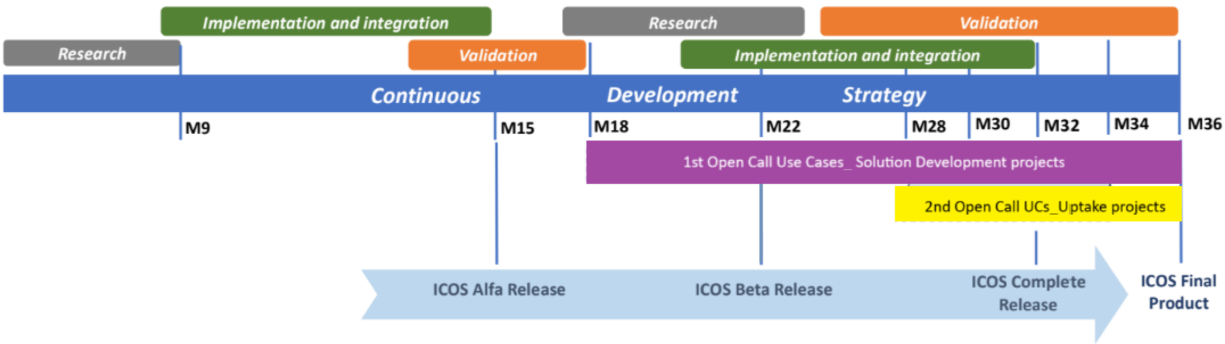

Figure 13 ICOS WP6 timeline

7 References

[1] **UC1: ICOS Agriculture Operational Robotic Platform – Demonstration VIDEO:** https://www.youtube.com/watch?v=t8-Mjz0m0is

[2] Based on "Nawożenie startowe jako metoda poprawy efektywności nawożenia i obniżki kosztów produkcji kukurydzy" by Michalski & Kowalik (Inżynieria Rolnicza, 2007), starter fertilization reduces overall doses and improves nutrient uptake efficiency

[3] **UC2: ICOS Railway Structural Alert Monitoring System – Demonstration VIDEO:** https://www.youtube.com/watch?v=X5Nx-8tUBgw

[4] **UC3: ICOS In-car Advanced Infotainment and Multimedia Management system – Demonstration VIDEO:** https://www.youtube.com/watch?v=5gPEhqTDeww

[5] ITU-T Recommendation E.800, Definitions of Quality of Service (QoS) parameters, International Telecommunication Union, 2008.

[6] ETSI GS NFV-TST 001, Network Functions Virtualisation (NFV); Pre-deployment Testing; Report on Proof of Concepts (PoC), European Telecommunications Standards Institute, 2014.

[7] Mahmud, R., Kotagiri, R., Buyya, R., Fog Computing: A Comprehensive Review, in arXiv preprint arXiv:1701.05451, 2018.

[8] Sarkar, S., Misra, S., "Theoretical modelling of fog computing: latency and cost analysis," in IEEE Transactions on Cloud Computing, vol. 6, no. 2, 2018.

[9] Hugging Face, Text Generation Inference Benchmarking, 2023. Available at: https://huggingface.co/blog/tgi-benchmarking

[10] Hugging Face, Text Generation Inference – Metrics Reference, 2023. Available at: https://huggingface.co/docs/text-generation-inference/en/reference/metrics

[11] **UC4: Energy Management and Decision Support system - Demonstration VIDEO:** https://www.youtube.com/@icos_project

[12] ICOS. D5.3 - Third ICOS Release: Complete ICOS version. A.Giannopoulos, M. Zetas, 2025 https://icos-project.eu/files/deliverables/D5.3_Third_ICOS_Release_v1.0.pdf

[13] ICOS. D2.2 – "ICOS Architectural Design (IT-1)", Gabriele Giammatteo, 2023. https://icos-project.eu/files/deliverables/D2.2_ICOS_Design_v1.0.pdf

[14] ICOS. D2.4 - ICOS architectural design (IT-2). J. Garcia, 2024. https://icos-project.eu/files/deliverables/D2.4-architectural-design-it2.pdf

[15] ICOS. D3.3-Meta-Kernel Layer Module Integrated IT-2. K. Meth, 2025. https://icos-project.eu/files/deliverables/D3.3-Meta-Kernel_Layer_Module_Integrated_IT-2.pdf

[16] ICOS. D4.2 - Data management, Intelligence and Security Layers (IT-2). AL. Suárez-Cetrulo, 2025. https://icos-project.eu/files/deliverables/D4.2_Data_Management_Intelligence_Security_Layers_IT-2.pdf