This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101069732

D5.4 – Use cases deployment and implementation (2)

Deliverable No.	D5.4	Due Date	31-JUL-2025 (submitted late: 9-SEP-2025)
Type	Report	Dissemination Level	Public
Version	1.0	WP	WP5
Description	*Will document the five pilots and integrated third parties, addressing the proposed use cases and scenarios. This will be the final version.*		

Copyright

The aerOS consortium consists of the following 27 partners:

UNIVERSITAT POLITÈCNICA DE VALÈNCIA	ES
NATIONAL CENTER FOR SCIENTIFIC RESEARCH "DEMOKRITOS"	EL
ASOCIACION DE EMPRESAS TECNOLOGICAS INNOVALIA	ES
TTCONTROL GMBH	AT
TTTECH COMPUTERTECHNIK AG (*third linked party*)	AT
SIEMENS AKTIENGESELLSCHAFT	DE
FIWARE FOUNDATION EV	DE
TELEFONICA INVESTIGACION Y DESARROLLO SA	ES
ORGANISMOS TILEPIKOINONION TIS ELLADOS OTE AE - HELLENIC TELECOMMUNICATIONS ORGANIZATION SA	EL
EIGHT BELLS LTD	CY
INQBIT INNOVATIONS SRL	RO
FOGUS INNOVATIONS & SERVICES P.C.	EL
L.M. ERICSSON LIMITED	IE
SYSTEMS RESEARCH INSTITUTE OF THE POLISH ACADEMY OF SCIENCES IBS PAN	PL
ICTFICIAL OY	FI
INFOLYSIS P.C.	EL
PRODEVELOP SL	ES
EUROGATE CONTAINER TERMINAL LIMASSOL LIMITED	CY
TECHNOLOGIKO PANEPISTIMIO KYPROU	CY
DS TECH SRL	IT
GRUPO S 21SEC GESTION SA	ES
JOHN DEERE GMBH & CO. KG*JD	DE
CLOUDFERRO SP ZOO	PL
ELECTRUM SP ZOO	PL
POLITECNICO DI MILANO	IT
MADE SCARL	IT
NAVARRA DE SERVICIOS Y TECNOLOGIAS SA	ES
SWITZERLAND INNOVATION PARK BIEL/BIENNE AG	CH

Disclaimer

Authors

Name	Partner	e-mail
Ignacio Lacalle	P01 UPV	iglaub@upv.es
Vasilis Pitsilis	P02 NCSRD	vpitsilis@iit.demokritos.gr
Andreas Sakellaropoulos	P02 NCSRD	asakellaropoulos@iit.demokritos.gr
Roger Briz Oscar Lazaro	P03 INNOVALIA	rbriz@innovalia.org olazaro@innovalia.org
Francesca Flamigni	P04 TTC	francesca.flamigni@tttech.com
Florian Gramss	P06 SIEMENS	florian.gramss@siemens.com
Fofy Setaki	P09 COSMOTE	fsetaki@cosmote.gr
Nikos Passas	P11 FOGUS	passas@fogus.gr
Katerina Giannopoulou	P11 FOGUS	kgiannopoulou@fogus.gr
Alex Kakyris	P11 FOGUS	akakyris@fogus.gr
Christos Milaroksotas	P11 FOGUS	milarokostas@fogus.gr
Paweł Szmeja	P13 SRIPAS	pawel.szmeja@ibspan.waw.pl
Vaios Koumaras	P15 INF	vkoumaras@infolysis.gr
Eduardo Garro	P16 PRODEVELOP	egarro@prodevelop.es
Álvaro Martínez Romero	P16 PRODEVELOP	amromero@prodevelop.es
Adrián Ramos	P16 PRODEVELOP	aramos@prodevelop.es
Juan Antonio Pavón	P16 PRODEVELOP	jpavon@prodevelop.es
Alessandro Cassera	P17 ECTL	Alessandro.cassera@eurogate-limassol.com
Kyriacos Orphanides	P17 ECTL	Kyriacos.orphanides@eurogate-limassol.com
Sheraz Aslam	P18 CUT	sheraz.aslam@cut.ac.cy
Herodotos Herodotou	P18 CUT	herodotos.herodotou@cut.ac.cy
Michalis Michaelides	P18 CUT	michalis.michaelides@cut.ac.cy
Sotiris Vasileiadis	P18 CUT	sr.vasileiadis@edu.cut.ac.cy
-	P21 JD	
Karol Klimaszewski Piotr Kłos Beata Kroszka	P22 CF	kklimaszewski@cloudferro.com pklos@cloudferro.com bkroszka@cloudferro.com
Bartosz Arendarski Tomasz Mrowiec	P23 ELECT	tmrowiec@electrum.pl barendarski@electrum.pl
Danish Abbad Syed	P24 POLIMI	danishabbas.syed@polimi.it

-	P25 MADE	
Daniel Cobo	P26 NASERTIC	dcobobor@nasertic.es
Lucie Stutz Sven Wallau	P27 SIPBB	lucie.stutz@sipbb.ch sven.wallau@sipbb.ch

History

Date	Version	Change
26-05-2025	0.1	Final Table of Contents
30-06-2025	0.2	First round of contributions
07-07-2025	0.5	Merged document with first round of contributions. Start final round of contributions
18-0-2025	0.6	Collection of final round of contributions.
31-07-2025	0.6	Merged document for Internal Review
12-08-2025	0.7	Internal Review performed and comments for improvement submitted
22-08-2025	0.8	Contributions by partners after internal review sent
31-08-2025	0.85	Final contributions integrated
05-09-2025	0.9	Submission to Project Coordination
09-09-2025	1.0	Final version for submission

Key Data

Keywords	IoT, Pilots, Quality Control, AGVs, Ports, Smart buildings, Renewable energy
Lead Editor	P03 INNO – Roger Briz
Internal Reviewer(s)	P11 FOGUS, P12 LMI

Executive Summary

This deliverable reports the second deployment phase of aerOS (M19–M35), documenting the end-to-end integration of the Meta-OS across five pilots (with four sub-pilots in Pilot 1) spanning manufacturing, energy, mobility, ports, and smart buildings. It extends D5.3's early setup work, focusing on how aerOS services were installed, configured, and operated in realistic environments, and sets the stage for validation and performance assessment to be reported in D5.6 at project end (M38).

Pilots have demonstrated how **aerOS has added value and improved their processes**, and the following paragraphs describe in a nutshell the advances of all pilots since M19

Pilot 1.1 — **Green manufacturing (zero net-energy) and CO2 footprint monitoring**

The SIPBB "Lighthouse" drone line was reconfigured in early 2025 (several stations rebuilt), which affected the original asset plan and shifted effort toward integration on the updated line.

Architecture actually deployed. A Raspberry Pi (Node-RED) collects station data at the edge; a ProLiant MicroServer hosts two VMs forming a Kubernetes cluster where aerOS services run; Orion-LD receives the station data. The Edge-vs-Cloud table shows the move from manual API pulls to managed services on K8s with edge data collection.

Stations & dataflows used. 3D Printer Farm, Smart Conveyor, Quality Check, Packaging, and the SMC Air Management System were integrated into the pipeline. Figures document Node-RED ingestion, Orion-LD state, and long-term storage (Prometheus/PostgreSQL).

Dashboards & outcomes. Dashboards visualize per-order carbon footprint and energy consumption; aerOS "basic" and "non-basic" components are listed as installed for this pilot.

Digital Product Passport (DPP) and compliance activities. The team built a Digital Product Passport data trail: component classification (supplied vs in-house), logistics footprint estimation, packaging, and total footprint calculation for each drone. They also experimented with the Gaia-X Wizard for credential issuance/signing.

Pilot 1.2 — **Automotive smart factory zero-defect manufacturing**

Deployment path. First a PoC with Entry Domain services at NASERTIC (identity/access control + secure connectivity), then full migration to Innovalia; final deployment at AIC with multiple CMMs operating in a live metrology setup.

The final architecture comprises three domains: (1) Entry (Keycloak/OpenLDAP, gateway), later moved to Innovalia; (2) Innovalia Metrology domain with CMM-adjacent edge IEs; and (3) M3 Software domain for metrology operations. Communication uses the OPC-UA RobotLink Server; CMMs are linked to the edge infrastructure (no direct HLO on machines).

Standardized machine access. A dockerized RLOPC service exposes machine parameters and methods over OPC-UA for aerOS/M3/HMI—documented screens list real-time attributes (positions, speeds, probe angles, energy values).

aerOS components and Ops. Screens show the installed aerOS services per domain (Entry and Metrology) and the Management Portal views (users, domains, continuum map).

Validated capabilities. Remote configuration/operation of CMMs, Digital Twin assembly for monitoring/early deviation detection, and dynamic execution of measurement services. The Edge-vs-Cloud table documents the shift of execution/control from cloud to edge with Self* modules and OPC-UA

Pilot 1.3 — **Zero Ramp-up safe PLC reconfiguration for Lot-Size-1 Production**

Three-tier setup online. Nuremberg runs the physical line (4 AGVs, mobile robot arm, safety door) with edge processing; Munich hosts a TSN lab; a restricted cloud tier runs orchestration only, all bridged with secure VPNs. Industrial Edge services now execute vision and data collection locally on SIMATIC IPCs, in line with Siemens' data policies.

Shop-floor network and safety stack in place. Dedicated TechHall subnet with SCALANCE firewall/topology; SICK NanoScan3 scanners + Safety PLCs integrated; AGV fleet manager configured with maps/paths.

Operational low-code orchestration. A behavior-tree (BT) workflow drives orders → asset relocation → sorting → opportunistic charging; aerOS handles container lifecycle (deploy/start/stop) for skills like navigation, lift, safety, ROS–TIA bridge.

TSN lab and zero-trust connectivity. SoCe MTSN switch with IPC + Raspberry Pi nodes; OpenZiti overlay links the TSN lab and primary aerOS domain; management/federation deployed (basic + non-basic components; Docker/NATS LLOs).

Federated secondary domain. NASERTIC configured as secondary for resilience/data-residency; Federator sync via OpenZiti entrypoints.

Pilot 1.4 — **AGV Swarm Zero break-down logistics for Lot-Size-1 Production**

Multi-domain pilot stood up. aerOS entry/management visible across the pilot continuum; network and domain infrastructure prepared (MADE dedicated network; single-node k3s + Raspberry Pi 5 at MADE/POLIMI).

Order-to-execution flow integrated. Order-manager apps (MADE & POLIMI) containerized with dual-arch images and registered in aerOS; NGSI-LD entities in Orion-LD enable order state and event sharing; a synthetic order generator supports end-to-end tests.

AGV navigation stack validated. ROS Noetic with AMCL localization, global planner, RPLIDAR, RViz monitoring; web AGV-Commander + Flask API bridge to ROS; end-to-end test from order creation to AGV mission confirmed.

Pilot 2 — **Containerised Edge Computing near Renewable Energy Sources**

Dual-site IE deployment. Two edge nodes on-prem with Kubernetes; central management on CloudFerro. Electrum's SCADA/IoT backend is connected; PV and RDHx cooling telemetry streams into Orion-LD.

Energy-price forecasting pipeline running. tgescrapper (ingest), price_prediction (model), and data_connector (publish) are containerized and orchestrated; models retrain/roll periodically to improve accuracy.

Right-sized placement. Computationally heavy cloud-mask processing kept in the cloud to balance CPU/GPU, while price forecasting, PV, and RDHx analytics execute at the edge.

LLO improvements and security. LLO patched for nodepool IE selection; semantic annotator/translator removed in favor of Data-Fabric connectors; (Kata) runtimeClassName limitations noted for future hardening.

Pilot 3 — **High Performance Computing Platform for Connected and Cooperative Mobile Machinery to improve CO2 footprint**

Vehicle and controller stack in place. SESAM electric tractor (JD) and R975i sprayer integrated; AutoTrac RTK guidance operational; TTControl Motion Board with Jetson AGX Xavier runs perception/control. Lab testbed is prepared for KPI runs and field validation windows (seasonal).

AI models have been deployed to achieve CO2 reduction via implementing a swarm orchestration of tractors. It includes integrating AI Models for field operation and orchestration (using AI for weed detection) and also the simulation of data orchestration for CO2 reduction (by integrating low latency networks), obtaining pre-measurements that will be validated in the on-field tests (September 2025).

Pilot 4 — **Smart edge services for the Port Continuum**

Predictive maintenance live. Data acquisition from Siemens S7-1500 (STS) and Omron CS1G (straddle carriers) via Siemens IoT2050 gateways (Node-RED); 4G backhaul; EUROGATE domain server aggregates and orchestrates; EntryPoint moved to AWS and linked by site-to-site VPN.

Visual inspection pipeline. Large, labeled datasets for container ID/damage/seal; dashboards for damaged-container evidence; Jetson Orin IE nodes onboarded; aerOS core services deployed at CUT and on Jetsons; inference/storage validated. Regarding Continuum integration & security, Port entrypoint and domains visible in the management portal; EAT and WireGuard overlay configured to traverse CG-NAT and secure east-west traffic.

Pilot 5 — **Energy Efficient, Health Safe & Sustainable Smart Buildings**

End-to-end smart-building loop closed. Two aerOS domains (Entry + Main) with KubeEdge at far-edge; IoT backend and HomeAssistant containerized; actuator controls HVAC/air-purifier/dehumidifier from optimization targets.

AI + Optimization + Recommender integrated. Health Index and Environmental Forecasting (XGBoost) publish to Orion-LD; an Energy-Efficiency regressor complements the stack; an optimizer computes target room conditions; a rule-based desk Recommender (Docker+Helm) serves the GUI over MQTT/NGSI-LD.

Data Fabric as the backbone. 11 IoT data products defined; Forecasting, Optimizer, Recommender, and GUI are all wired through the Data Fabric/Orion-LD; real-time user flows (presence → recommendation) demonstrated.

5G security extensions. OpenCAPIF deployed to expose NEF securely; UPF VNF onboarded in the continuum and validated with UERANSIM; RBAC enforced via LDAP/Keycloak/KrakenD.

aerOS platform outcomes: Across pilots, aerOS orchestration is now exercised consistently: HLO (allocation, deployment, data aggregation), LLO-K8s, identity and API gateway services (Keycloak, KrakenD, OpenLDAP), the Federator, Orion-LD, and self-* modules are deployed broadly; LLO-Docker complements mixed infrastructures where not all assets run Kubernetes. This coverage is summarized in the Components Assessment (Table 29).

Regarding Standards & compliance.. Pilots adopt domain standards that make deployments replicable: NGSI-LD/NGSI-LD API (context information), OPC UA (device/PLC integration), IEEE TSN (deterministic networking), PROFINET/PROFIsafe (industrial communication & safety), DDS (ROS 2 middleware), and TIC 4.0 (CHE semantics). GDPR compliance was assessed via a structured checklist; pilots 1.1, 1.2, 1.3, 2, and 4 reported no personal-data processing, with documentation retained in the project repository.

The Open Calls have yielded relevant outcomes. OC#1's feedback directly improved installation guides, identity configuration, portal features (private registries, Helm customization), and packaging validations (LLO charts; aeros-k8s-shim decoupling). OC#2 –just finalized—funded eight projects; support focused on troubleshooting installation/customization in new verticals.

D5.4 evidences operational readiness of aerOS across heterogeneous, safety-critical, and data-sensitive domains, with orchestration, identity, federation, and semantics functioning cohesively at scale. Formal validation and KPI results will follow in D5.6 (M38).

Table of contents

List of tables

List of figures

List of acronyms

Acronym	Explanation	Acronym	Explanation
AI	Artificial Intelligence	**K8S**	Kubernetes
AGV	Autonomous Ground Vehicle	**LAN**	Local Area Network
AMR	Autonomous Mobile Robot	**LLO**	Low Level Orchestrator
API	Application Programming Interface	**ML**	Machine Learning
CD	Continuous Development	**MVP**	Minimum Viable Product
CH	Chapter	**MQTT**	Message Queuing Telemetry Transport
CI	Continuous Integration	**OPC**	Open Platform Communications
CMM	Coordinate-measuring machine	**OT**	Operational Technology
DPP	Digital Product Passport	**RAM**	Random Access Memory
DT	Digital Twin	**RIB**	Realtime Information Backbone
ECU	Electronic Control Unit	**ROS**	Robot Operating System
EU	European Union	**RTO**	Research and Technology Organisations
ELM	Evocortex Localization Module	**TH**	Trial Handbook
ERP	Enterprise Resource Planning	**SIM**	Subscriber Identity Module
ESR	Evaluation Summary Report	**SME**	Small and Medium-sized Enterprises
GHZ	Gigahertz	**SSF**	Swiss Smart Factory
GNSS	Global Navigation Satellite System	**PCB**	Printed Circuit Board
GPU	Graphics Processing Unit	**PC**	Personal Computer
HLO	High Level Orchestrator	**PCC**	Project Coordination Committee
HW	Hardware	**PLC**	Programmable-Logic Controller
IoT	Internet of Things	**PLM**	Product Lifecycle Management
IE	Infrastructure Element	**PoC**	Proof of Concept
IP	Internet Protocol	**VM**	Virtual Machine
IPC	Industrial Personal Computer	**VPN**	Virtual Private Network
IT	Information Technology	**WIFI**	Wireless Fidelity
KPI	Key Performance Indicator	**WIP**	Work In Process

1. About this document

This document presents a consolidated report on the implementation and integration of the aerOS pilots and third-party use cases during the second deployment phase of the project, covering the period from M19 to M35. It follows and expands upon the initial reporting provided in deliverable D5.3, which focused on setup and early-stage integration activities across the pilot sites.

D5.4 describes the full technological deployment of the aerOS MetaOS across all five pilots—including the 4 subpilots of pilot 1—detailing how its components have been integrated into real industrial, energy, mobility, logistics, and smart building environments. The document captures the evolution of each pilot's infrastructure and the application of aerOS services within operational contexts, with a particular focus on orchestration, data autonomy, trust, security, and the management of the edge-cloud continuum.

Each pilot section includes a description of the demonstrator path, the final technical architecture adopted, and the main setup, development, and integration activities carried out. An updated assessment of the aerOS components deployed is also included, along with references to standards adopted and any relevant input from Open Call experiments.

Validation activities and associated performance assessments will be documented separately in Deliverable D5.6, which will be submitted at the end of the project by M38.

This deliverable serves as a key milestone in demonstrating the operational readiness and cross-domain applicability of aerOS, reflecting the maturity and adaptability of the platform across a diverse set of real-world use cases.

1.1. Deliverable context

This deliverable contributes directly to the objectives of WP5 – Integration, deployment, and validation of aerOS in real-life pilots, with a particular focus on the deployment and integration phases of the aerOS MetaOS across all use cases

Item	Description
Objectives	**O1 (Design, implementation and validation of aerOS for optimal orchestration)**. Operationalizing the design and validation aspects of aerOS, ensuring optimal orchestration in real-world scenarios. This deployment phase transforms the theoretical framework of aerOS into practical applications **O2 (Intelligent realisation of smart network functions for aerOS).** The deployment outcomes contribute valuable insights, validating the aerOS network capabilities. **O3 (Definition and implementation of decentralised security, privacy and trust)**. The practical application of aerOS in use cases significantly advances the objectives set forth for security, privacy, and trust within the aerOS project. **O5 (Specification and implementation of a Data Autonomy strategy for the IoT edge-cloud continuum).** Through practical implementation, aerOS becomes a robust supporter of the diverse needs of data producers and consumers. **O6 (Definition, deployment, and evaluation of real-life use cases)**. The deployment ensures the utilization of real-world data, considering factors such as interoperability, volume, variety, and rate, to validate the platform's performance and business impact.
Work plan	D5.4 Receives inputs from: **WP2** (reference architecture for the IoT edge-cloud continuum): architectural guidelines, design principles, and key considerations specified by WP2 to ensure that the deployed use cases align seamlessly with the established architectural framework. **WP3** (secure, scalable and decentralised compute infrastructure): guide and provide the aerOS

	components for their implementation in the use cases. **WP4** (delivering applications intelligence at the edge): guide the implementation of intelligent components within aerOS, ensuring that the platform effectively supports applications with enhanced intelligence capabilities. **WP5 (**aerOS Integration and use cases deployment): aerOS components integration and deployment on every scenario. <u>D5.4 Influences:</u> **WP5** (aerOS validation activities and KPIs): Will be reported in the upcoming D5.6 (M38).
Milestones	This deliverable contributes directly to the realisation of *MS5 - Final architecture defined: final architecture produced with feedback from the use cases* - and confirms the completion of *MS6 - Final integrated software solution* -: components of the aerOS system deployed and fully integrated within the pilot environment- and *MS7 - Final software components released: validated aerOS components running in production-like conditions*. It also provides ongoing contribution toward MS8 (Project results validated in pilots and final review preparation), with final validation activities planned for completion before the project's end by M38.
Deliverables	This deliverable is the second and final version of a two-part report, with the first version delivered at M18 and this version at M35 (July 2025) – *with 5 weeks delay*. It has received inputs from D2.1 (State-of-the-Art and market analysis report), D2.2 (Use cases manual, requirements, legal and regulatory analysis (1)), D2.6 (aerOS architecture definition (1)), D3.1 (Initial distributed compute infrastructure specification and implementation), D4.1 (Software for delivering intelligence at the edge – preliminary release), and D5.1 (Integration, evaluation plan and KPIs definition (1)). The results presented here will provide inputs and contribute to D5.6 (Technical evaluation, validation and assessment report 2), to be delivered at the end of the project – M38.

By covering the period from M19 to M35, this deliverable reports on the second and most advanced stage of deployment, demonstrating how aerOS has evolved from component-level maturity into a fully integrated and operational framework applied in industrial and domain-specific environments.

The activities described in this report are essential for preparing the ground for the final validation phase of the project. The outcomes reported here feed directly into the work planned for Deliverable D5.6, which will focus on validation, KPIs, and impact assessments during the final project phase (up to M38).

1.2. Outcomes of the deliverable

This deliverable presents the second-stage outcomes of the aerOS pilot deployment, covering the period from M19 to M35 and capturing the transition from early infrastructure setup and integration—as reported in the previous deliverable D5.3—to the full-scale implementation and operational use of aerOS technologies across all pilots and sub-pilots.

The outcomes documented include:

- A comprehensive report on the deployment of the aerOS MetaOS on real operational environments across various industrial, energy, port logistics, and smart building domains.
- Detailed assessment of the aerOS components installed per pilot, highlighting capabilities for orchestration, data autonomy, self-management, and edge-cloud operation.
- Updated technical architectures and integration setup and workflows across all pilots.
- A consolidated view of the standards applied in each use case, focusing on those with direct technical or interoperability relevance.

- Outcomes of the integration activities with third-party partners from the first Open Calls, highlighting pilot-specific extensions and collaborations.
- An overview of GDPR-related considerations and pilot compliance with data protection principles.

These results lay the groundwork for the final phase of the project, which will focus on validation and impact assessment. These activities will be fully covered in D5.6, titled Validation and Assessment of the aerOS Pilots

1.3. Lessons learnt

The deployment of aerOS across the 5 pilots (and 4 subpilots of pilot 1) provided clear evidence of both achievements and challenges when operating a Meta-OS in operative environments. Pilots confirmed that aerOS can run reliably on heterogeneous infrastructure (and over varying container management frameworks), but also showed that careful adaptation is required. Kubernetes clusters, lightweight edge devices, and Docker-based nodes all proved viable, yet differences in capacity and integration effort demanded flexibility in packaging and deployment.

Another lesson is the value of placing computation as close as possible to the assets: by shifting analytics and monitoring functions to the edge, pilots improved responsiveness, resilience, and privacy, while keeping demanding tasks in the cloud. Standards such as OPC UA, NGSI-LD, TSN, and TIC 4.0 enabled interoperability with existing systems and were decisive in integrating brownfield equipment. Security and compliance mechanisms, including federated identity, access control, and network policies, were successfully applied in all pilots..

The Open Calls (OC) brought complementary lessons from external adopters. OC#1 revealed that installation and onboarding procedures were not always straightforward, leading to improvements in documentation, Helm charts, and component decoupling. OC#2, still ongoing, has shown the importance of supporting newcomers during setup and customization. It also highlighted that aerOS can be extended to new verticals, but only if the entry barriers are kept low and support mechanisms remain strong. These insights from the OCs demonstrate that while pilots validate aerOS in depth within defined domains, external users are equally valuable for testing usability, portability, and the attractiveness of the platform to a wider ecosystem

2. Progress of Use Cases deployment and implementation

This section presents a detailed update of the aerOS pilots' progression during the second deployment phase, covering activities carried out between M19 and M35. It builds on the structure established in D5.3, which introduced the validation approach and pilot activity categorization adopted by WP5.

The aerOS validation strategy remains structured around three phases: PoC (M1–M12), MVP and system releases validation (M12–M24), and system scaling and optimization (M24–M36). D5.4 documents the transition from the second to the third phase, where pilots consolidated their deployments and prepared their infrastructures and services for operational maturity.

Each pilot followed a common implementation path organized into four main activity types: Setup and Procurement, Development, Integration, and Validation. This categorization supports consistent tracking of progress across highly diverse use cases and domains.

In this deliverable, each pilot report includes an updated description of the demonstrator path, the final architecture adopted, and the results of the main setup, development, and integration tasks. Validation activities and performance measurements will be fully addressed in Deliverable D5.6, due by M38.

For additional details about the validation plan or activities categorization, please refer to sections 2.1 and 2.1.1 from the previous D5.3.

2.1. Pilot 1 – Data-Driven Cognitive Production Lines

Pilot 1 demonstrates the deployment of the aerOS MetaOS across four distinct but complementary industrial environments, each addressing different manufacturing challenges and technological contexts. The sub-pilots—**1.1 (SIPBB), 1.2 (INNOVALIA), 1.3 (SIEMENS), and 1.4 (MADE & POLIMI)**—serve as complementary use cases to test aerOS functionalities in production lines, metrology systems, automation networks, and flexible manufacturing setups.

These sub-pilots collectively contribute to the validation of Edge-Cloud orchestration, AI-assisted decision-making, self-management services, and the interoperability of industrial systems within aerOS. Each site has adapted its architecture and operational workflows to integrate aerOS components in a real-world production setting, enabling performance improvements in areas such as data acquisition, machine control, process optimization, and standard compliance.

Each of the following sections—2.1.1, 2.1.2, 2.1.3 and 2.1.4—describe the progress and outcomes of each sub-pilot during the second phase of implementation

2.1.1. Green manufacturing (zero net-energy) and CO_2 footprint monitoring

Scenario 1 takes place at the Swiss Smart Factory (SSF), in Switzerland Innovation Park Biel/Bienne (SIPBB). The Swiss Smart Factory main activity is the "Lighthouse Project Industry 4.0", which consists of a modular production line of drones. Its purpose is to demonstrate how new technologies can be operated within a factory, and it welcomes more than 3'000 visitors each year. Therefore, this production line evolves rapidly throughout time, depending on the new technologies available on the market. This can make it challenging to remain consistent on the long run regarding the technologies and assets available at the SIPBB during pilot deployment within innovation projects. Furthermore, SIPBB encountered an extraordinary activity at the start of 2025, due to another running project, where five of the main assets constituting the drone production line were definitely removed and had to be rebuilt internally by the SIPBB team. Some of these assets were originally identified to run aerOS pilot 1 scenario 1, creating a delay for integration activities, which will be further explained in the according sections below.

This deliverable departs from D5.3 at the point where the setup, procurement, and initial architecture were established. This included the identification of the four pilot stations: Packaging, Arm Printing, Smart

Conveyor, and Quality Check. The hardware infrastructure was defined, comprising Raspberry Pi devices running Node-RED to collect data from the stations, as well as a ProLiant MicroServer hosting two virtual machines to form a Kubernetes cluster for aerOS deployment. Furthermore, the IT architecture and data flow were outlined, ensuring seamless communication from the assets to Orion-LD for data aggregation, storage, and subsequent analysis. The validation process, including a time plan and activity table, was set to guide the project's next steps. All of these activities ensured that the pilot environment was properly structured and aligned with the broader goals of the aerOS project.

The focus now shifts towards integration. Specifically, it outlines the progress made in deploying aerOS services within the SIPBB domain, integrating the collected data from production stations, implementing dashboard visualizations. The activities described in this deliverable build upon the established infrastructure and provide detailed insight into the operationalization of the pilot system.

Since last year, the hardware and software required to integrate aerOS components into the pilot have been finalized. The pilot tracks and analyses energy and CO2data from machines in a drone production line. Functions for collection, analysis, storage, and visualization of energy/CO2data are implemented with the Embedded Analytics Tool (EAT) integrated into aerOS: (i) collection from Orion-LD, long-term storage in Prometheus, and visualization in Grafana; (ii) calculation of total CO2per asset and per drone order with results written to PostgreSQL; and (iii) generation of forward-looking CO2estimates using an ARIMA model trained on historical data. The Deployed OpenFaaS function are shown in the screenshot below. Then, a Digital Product Passport (DPP) was implemented and contributes to greener manufacturing by enabling local, edge-driven analytics on the SSF production line

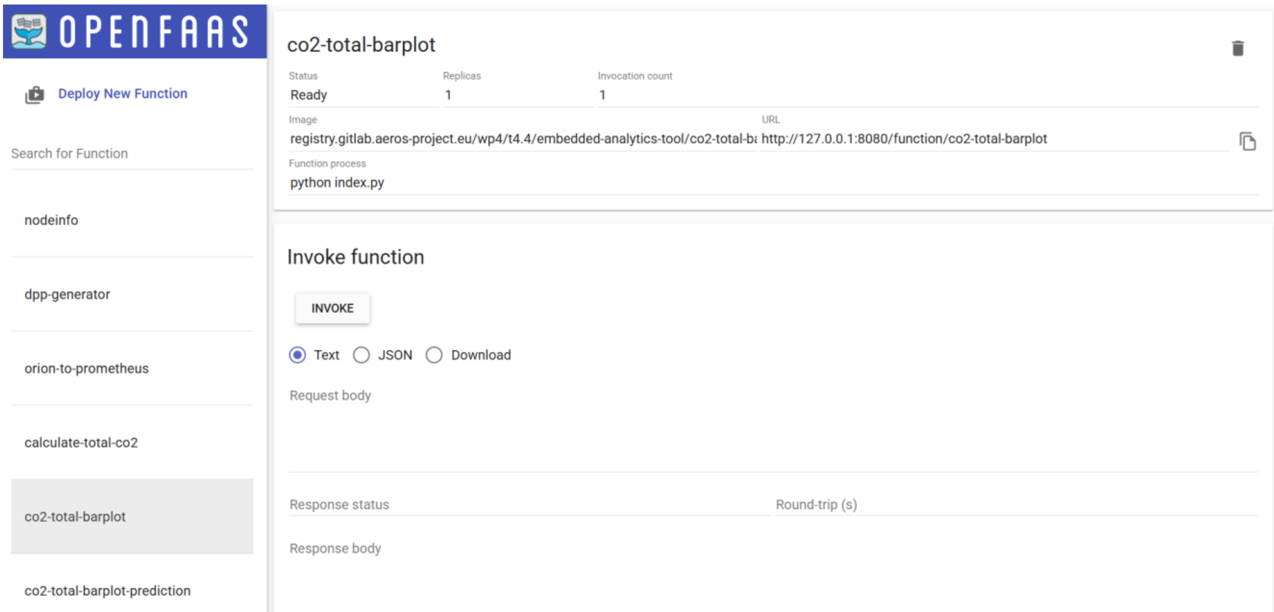


Figure 1. Pilot 1.1 Deployed OpenFaas function for collection, analysis, storage, and visualization of energy/CO2data

```
vm1@vm1-VirtualBox:~$ kubectl get pods -n openfaas-fn
NAME                                            READY   STATUS    RESTARTS       AGE
calculate-total-co2-79d9dfbcb4-vgpz7            1/1     Running   80 (20h ago)   39d
co2-total-barplot-5dc758ddb8-54zdh              1/1     Running   0              19h
co2-total-barplot-prediction-58dbcfb857-qhm62   1/1     Running   0              19h
dpp-generator-6cfdcdd9bb-vhwd8                  1/1     Running   0              19h
nodeinfo-77c6756589-nxhns                       1/1     Running   60 (21h ago)   161d
orion-to-prometheus-58b7dbc6d-6cpxg             1/1     Running   0              20h
```

Figure 2. Pilot 1.1 Pilot 1.1 Deployed OpenFaas function running inside the Kubernetes cluster

CO_2: The integration into the aerOS portal and the deployment of the services across the continuum was delayed due to network security concerns from SIPBB that first had to be clarified, but this process was finalized successfully.

Additionally, experimentations with the Gaia-X Wizard were made to gain practical understanding of credential issuance and signing using Verifiable Credentials (VCs), which are used to prove Gaia-X compliance.

The following table showcases how pilot 1.1 has switched from processing data in the Cloud to the Edge:

Table 1. Pilot 1.1 Edge vs Cloud comparison before and after aerOS.

	Cloud	Edge
Before aerOS	No data processing in the cloud. Data requests were made manually using the Stations REST API.	No data processing at the edge. Data requests were made manually using the Stations REST API.
After aerOS	Data storage, processing, and analytics are performed using aerOS components managed by Kubernetes.	Data collection is performed using a Raspberry Pi as an edge device.

2.1.1.1. Technical schema

The diagram illustrates the architecture of pilot 1 scenario 1 (aerOS SIPBB Domain), which consists of three Infrastructure Elements (IE) and the aerOS NASERTIC Domain.

IE1 and IE2 are two virtual machines (VM1 and VM2) running on the ProLiant MicroServer.

They form the Kubernetes cluster that hosts and manages all aerOS components (represented by the colored rectangles within the blue IE1/IE2 box) and the related software.

IE3 is a Raspberry Pi running a dockerized Node-RED instance. It collects data from all connected assets and sends it to the Orion-LD Context Broker, which is part of the aerOS software stack running as a pod within the Kubernetes cluster.

This architecture was chosen for its practicality: the Raspberry Pi efficiently collects real-time data at the edge, while the ProLiant server with its two VMs provides a stable environment to operate the Kubernetes cluster and ensure reliable deployment of all aerOS services.

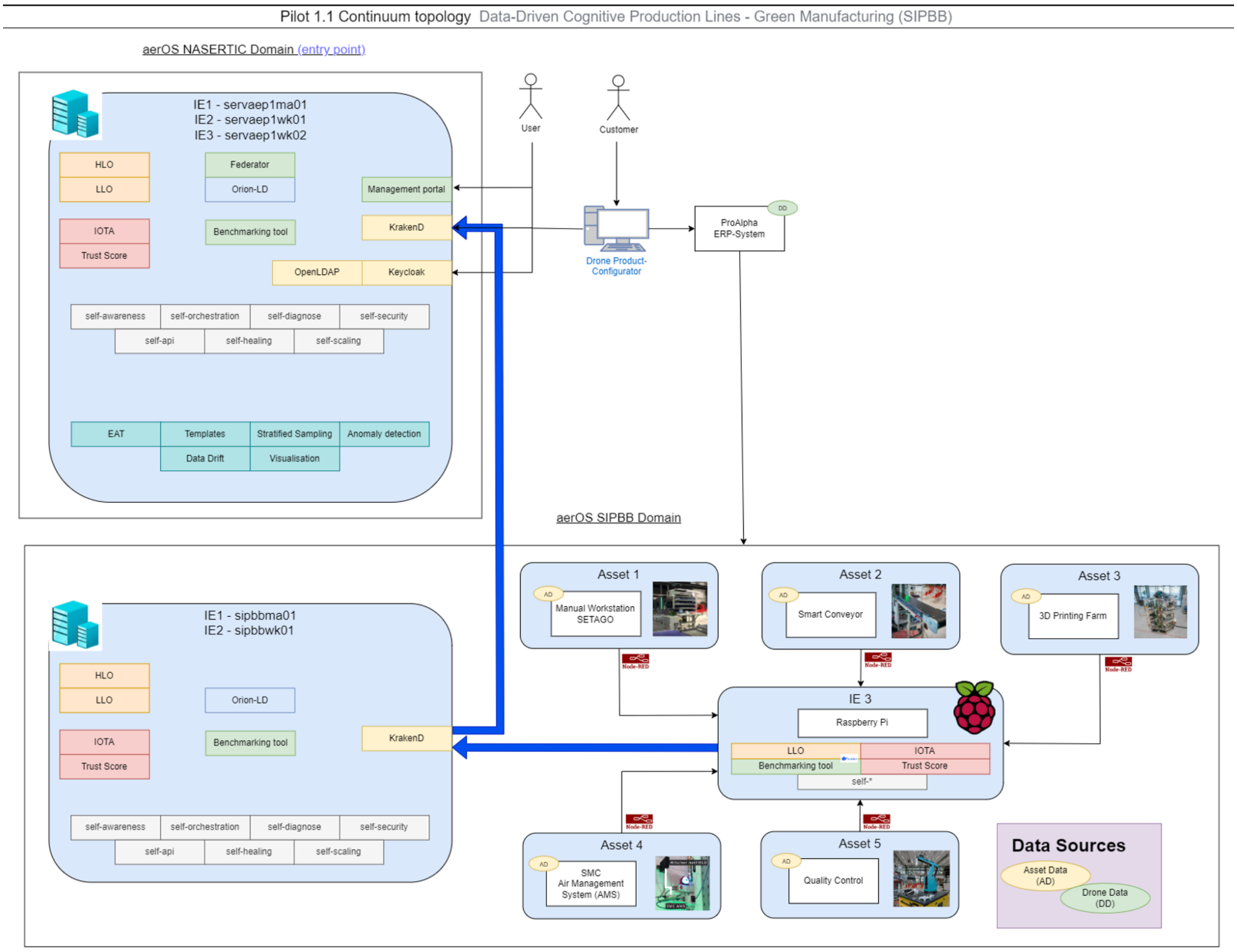


Figure 3. Pilot 1.1 Topology diagram and technical schema of components deployed

2.1.1.2. Report of activities

2.1.1.2.1. Setup and procurement activities

Pilot1 – Business Process 1 – Activity - 1 (P1-BP1-SA1): Stations identification for the trial

In the initial setup phase, some of the production stations that make up the drone manufacturing line in the Swiss Smart Factory were selected to be part of the aerOS project. These are the physical work cells and equipment needed for data collection.In D5.3, the stations originally identified were: manual workstation Setago, smart conveyor, PCB THT-Assembly, Melkus C4060 AGV and quality check station. Due to parallel projects and activities at SIPBB mentioned in the introduction, some adjustments had to be made. During the first part of aerOS deployment, the following machines were used:

1. 3D Printer Farm: consisting of nine 3D printers, this farm produces the arms of the drone. Each printer creates a different color. When a customer orders a drone, they have to choose its configuration, including defining the model and the colors they want. Once completed, the configuration data is sent to the company's ERP and to the machines involved as JSON format. Production automatically starts and begins with this station.

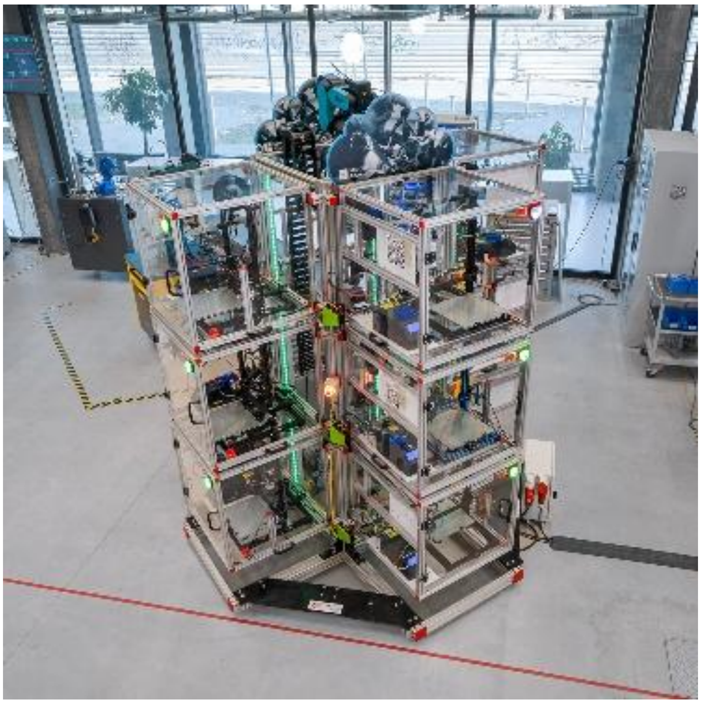


Figure 4. 3D Printer Farm (P1.1 SIPBB)

1. Smart Conveyor: once printed, the arms are transported from the 3D Printer Farm to the next station, thanks to the Smart Conveyor. QR-Codes on the conveyor itself act as parking slots for the arms, defining into the system to which drone belongs the arm.

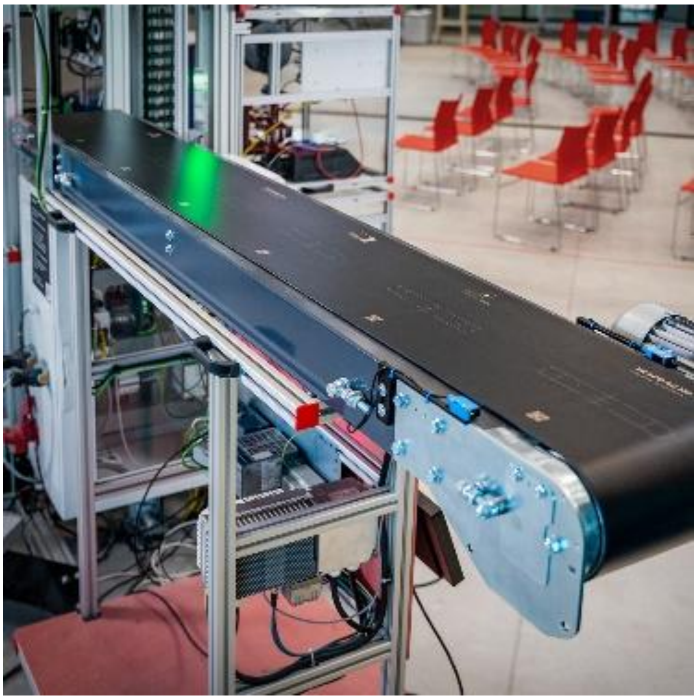


Figure 5. 3D Smart conveyor (P1.1 SIPBB)

2. Quality Check: a robot picks up the arm from the smart conveyor and proceeds to control the flatness of the arm, while also integrating an NFC cheap inside each drone.

Figure 6. Quality check (P1.1 SIPBB)

3. Packaging: after the drone has been assembled, an operator follows digital instructions to pack the drone and its side components before sending it to the warehouse. Digital instructions are used to facilitate the training of new employees and to ensure minimal mistakes are made due to the different configurations possible.

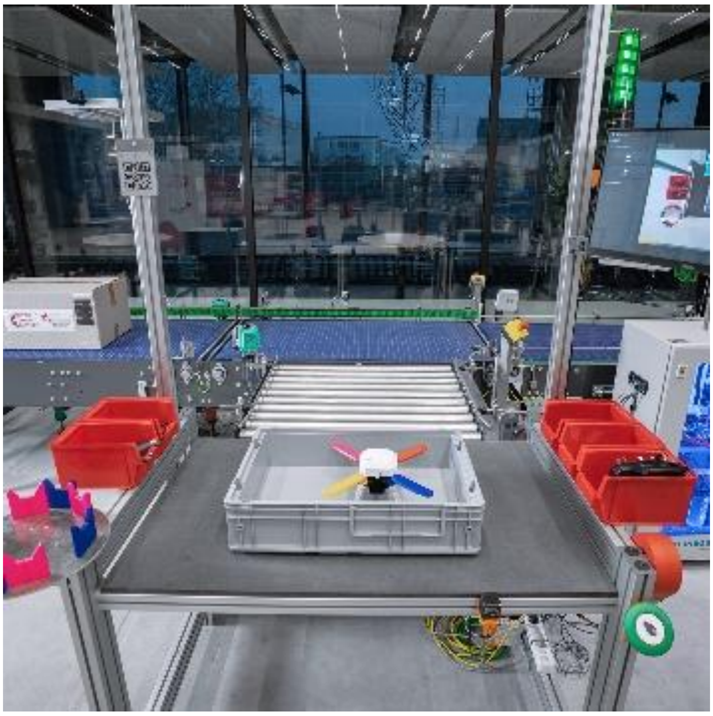


Figure 7. Packaging machine (P1.1 SIPBB)

4. SMC Air Management System (AMS): with SMC's Air Management System, Balluff's Smart Light, and Hilscher's netFIELD Compact Edge Device, connected via an OPC-UA server and Node-RED, it enables the factory to optimize airflow to machines, using it only when needed.

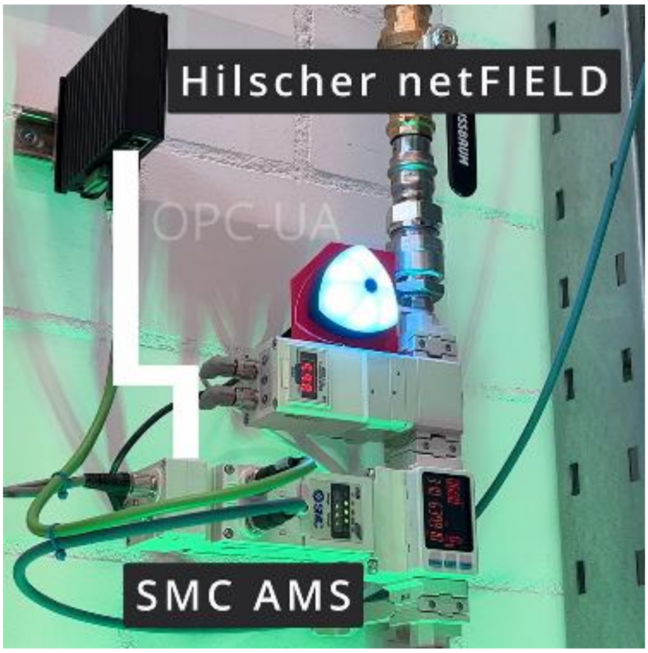


Figure 8. SMC Air Management System (P1.1 SIPBB)

Each asset provides data about its status, environmental data and energy monitoring including CO2 footprint emissions generated, as presented in the figure below.


```json
{
    "machine_status": {
        "machinename": "Quality_check",
        "status": 1,
        "orderid": 70012,
        "timestamp": 1731487994806,
        "specific_machine_data": []
    },
    "environmental_data": {
        "available_data": [
            "temperature",
            "humidity",
            "vibration",
            "atmospheric_pressure"
        ],
        "temperature": 26,
        "humidity": 35.8,
        "vibration": 0.0112525,
        "atmospheric_pressure": 978.1
    },
    "energy_monitoring": {
        "available_data": [
            "power_current",
            "power_cumulated",
            "co2_footprint_current",
            "co2_footprint_cumulated"
        ],
        "power_current": 885,
        "power_cumulated": 17.058,
        "co2_footprint_current": 52.215,
        "co2_footprint_cumulated": 2375.813
    }
}
```


Figure 9. Example of Quality Check Asset Data (P1.1 SIPBB)

For the final phase of the pilot including validation activities, some stations might differ. The potential impact on the trial will be described in D5.6.

Pilot1 – Business Process 1 – Activity - 2 (P1-BP1-SA2): Hardware setup

After identifying the stations, the hardware infrastructure necessary to support the aerOS pilot environment was set up. It focused on preparing the physical and computational elements required for local data processing and system integration.

Hardware setup:

- Raspberry Pi: Lightweight edge devices connected directly to the stations. These devices run a Dockerized Node-RED application, which:
    - Fetches real-time data from physical assets at each station.
    - Send the data to the Orion-LD Context Broker, enabling data analysis and monitoring within the aerOS ecosystem.
- ProLiant MicroServer: Acts as the main on-site server for orchestration and hosting:
    - Two VMs are deployed to form a Kubernetes cluster, consisting of a control plane and a worker node.
    - Core aerOS software components are deployed within this Kubernetes environment.

Pilot1 – Business Process 1 – Activity - 3 (P1-BP1-SA3): Equipment configuration

In this step, all factory assets and software were configured to ensure seamless integration. This involved assigning static IPs to each identified station (Packaging, Printing, Conveyor, Quality), as well as to the Raspberry Pi, the ProLiant MicroServer, and the virtual machines (VMs), in order to connect them to the network and enable communication between them. The VMs running on the ProLiant MicroServer had to be made accessible via a network bridge adapter, which can be configured inside the VirtualBox Manager, as shown in the image below.

Some services, such as Grafana, Prometheus, PostgreSQL, or the EAT Gateway, were running inside the Kubernetes cluster and had to be exposed via port forwarding to make them accessible outside the Kubernetes pods, allowing the EAT function to access these services. This also enabled the Grafana dashboards to be available from any browser within the network. Port forwarding was also necessary for the containerized Node-RED application running on the Raspberry Pi.

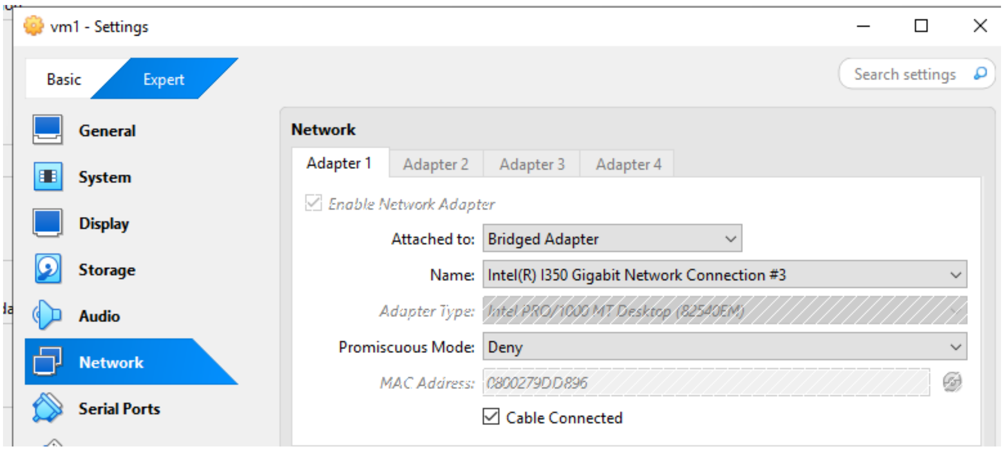


Figure 10. Pilot 1.1 Bridge Adapter Configuration for VM1

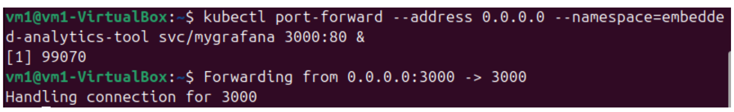


Figure 11. Pilot 1.1Port forwarding for the Grafana service

2.1.1.2.2. Development activities

Pilot1 – Business Process 1 – Activity - 4 (P1-BP1-DA4): Definition of IT architecture

All the infrastructure elements, Assets Raspberry PI and Server are connected to the factory Wi-Fi. The following illustration shows the data flow from the assets to the aerOS components:

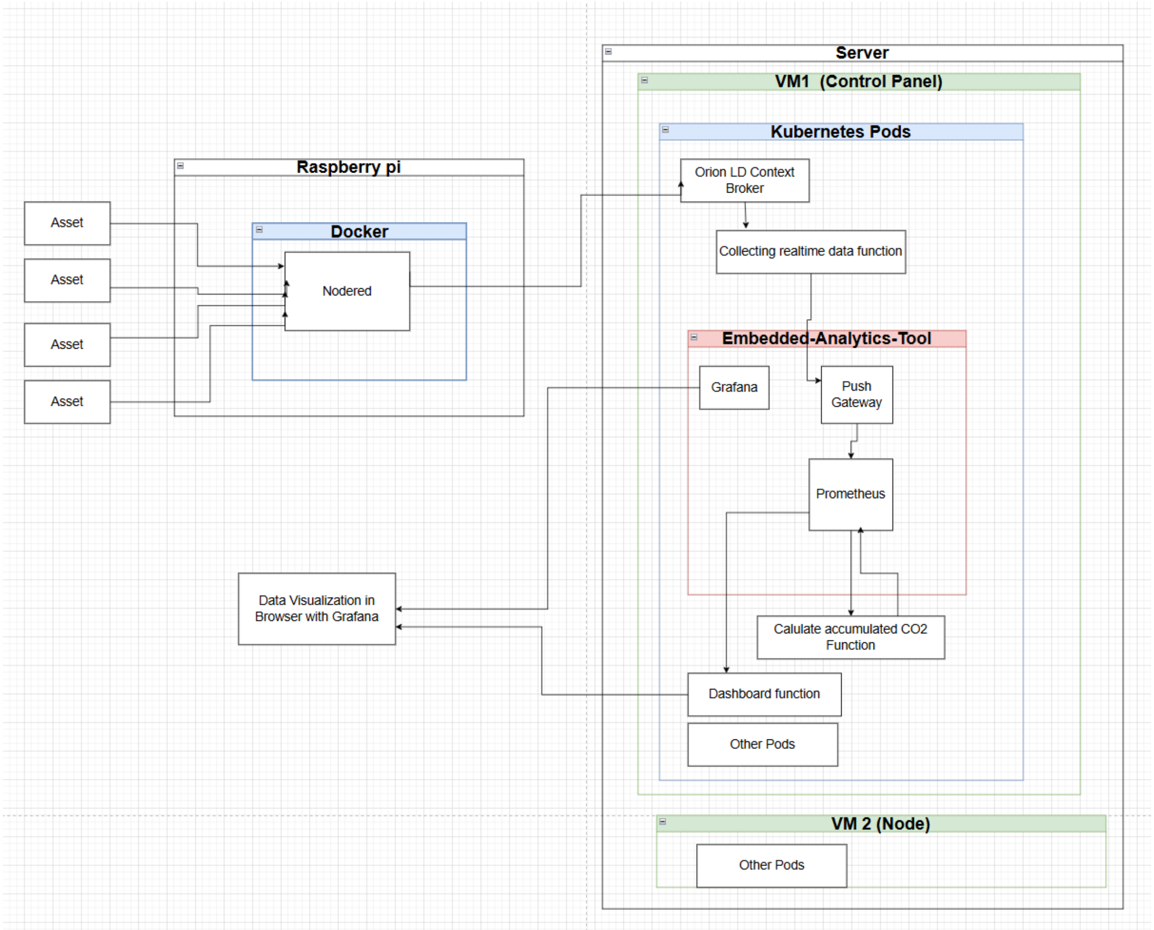


Figure 12. Pilot1.1. Data flows from assets to aerOS Meta OS software components

Pilot1 – Business Process 1 – Activity - 5 (P1-BP1-DA5): Communication infrastructure developed or adapted

The low-code tool Node-RED was successfully implemented, fetching data from the asset and sending it to Orion-LD. This communication does not need to flow through KrakenD since it is an intrinsic part of internal communication within aerOS components. There is no external connection at this point, therefore it keeps consistency with aerOS Meta-OS design

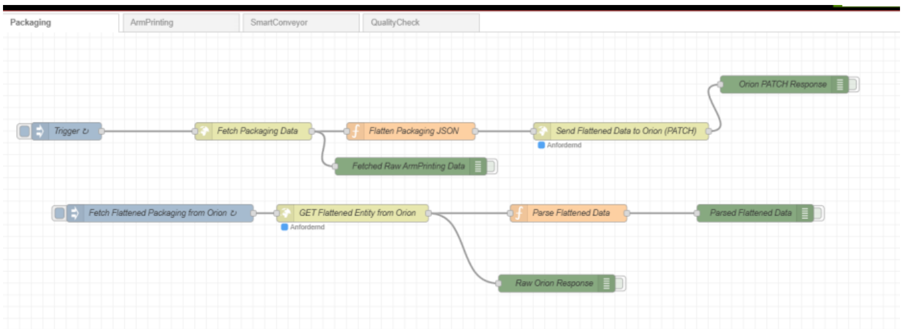


Figure 13. Pilot 1.1 Node-RED workflow for retrieving data from the packaging machine, converting it to NGSI-LD format, and transmitting it to Orion-LD

The data retrieved from the asset endpoints is provided in JSON format by the machines and includes energy-related information such as power consumption (current in W and cumulated in kWh) and CO_2 footprint (current in g and cumulated in kg). In addition, it contains environmental parameters measured internally by the machines, including temperature and humidity.


```
        "energy_monitoring": {
            "available_data": ["power_current", "power_cumulated", "co2_footprint_current", "co2_footprint_cumulated"],
            "co2_footprint_cumulated": 9182.682,
            "co2_footprint_current": 0.011,
            "power_cumulated": 4.536,
            "power_current": 9.15
        },
        "environmental_data": {
            "atmospheric_pressure": 961,
            "available_data": ["temperature", "humidity", "vibration", "atmospheric_pressure"],
            "humidity": 61.5,
            "temperature": 17.8,
            "vibration": 4.995
        },
        "machine_status": {
            "machinename": "Packaging",
            "orderid": "",
            "specific_machine_data": [],
            "status": 1,
            "timestamp": 1750323643119
        }
    }
```


Figure 14. Pilot 1.1 Raw Json data format coming from the Packaging machine

Within the Node-RED flow, the retrieved JSON data is transformed into NGSI-LD format, as required by Orion-LD for storing and managing the data

```
var data = msg.payload;

var isoTimestamp = "";
if (data.machine_status && data.machine_status.timestamp) {
    var ts = parseInt(data.machine_status.timestamp);
    isoTimestamp = new Date(ts).toISOString();
}

msg.payload = {
    "machinename": {
        "type": "Property",
        "value": data.machine_status ? data.machine_status.machinename : "Packaging"
    },
    "status": {
        "type": "Property",
        "value": data.machine_status ? data.machine_status.status : 1
    },
    "orderid": {
        "type": "Property",
        "value": data.machine_status.orderid
    },
    "timestamp": {
        "type": "Property",
        "value": isoTimestamp
    },

    // Environmental
    "temperature": {
        "type": "Property",
        "value": data.environmental_data ? data.environmental_data.temperature : 25
    },
    "humidity": {
        "type": "Property",
        "value": data.environmental_data ? data.environmental_data.humidity : 50
    },
    "vibration": {
        "type": "Property",
        "value": data.environmental_data ? data.environmental_data.vibration : 0.5
    },
    "atmosphericPressure": {
        "type": "Property",
        "value": data.environmental_data ? data.environmental_data.atmospheric_pressure : 1013
    },

    // Energy
    "power_current": {
        "type": "Property",
        "value": data.energy_monitoring ? data.energy_monitoring.power_current : 10
    },
    "power_cumulated": {
        "type": "Property",
        "value": data.energy_monitoring ? data.energy_monitoring.power_cumulated : 2.0
    },
    "co2_footprint_current": {
        "type": "Property",
        "value": data.energy_monitoring ? data.energy_monitoring.co2_footprint_current : 0.01
    },
    "co2_footprint_cumulated": {
        "type": "Property",
        "value": data.energy_monitoring ? data.energy_monitoring.co2_footprint_cumulated : 500
    }
};
return msg;
```

Figure 15. Pilot 1.1 Node-RED code for converting raw JSON data into the NGSI-LD format expected by Orion-LD

Orion-LD maintains the current state of the machine data and serves as a central hub for managing and distributing this information. For long-term storage, an Embedded Analytics Tool (EAT) function is deployed, which retrieves the current machine state from Orion-LD and stores it in both a Prometheus and PostgreSQL database. Evidence is shown in *P1-BP1-DA7*.

Pilot1 – Business Process 1 – Activity - 6 (P1-BP1-DA6): APIs setup

Each station/asset has its own endpoint with a pre-defined data structure.

Each machine's endpoint is triggered in Node-Red, converted into NGLS-LD format and sent to its Orion-LD Entity. The following images show how the machines endpoints are triggered in Node-RED, what the Json data looks like and how the data is transmitted to Orion-LD, according to the mechanisms defined in aerOS' Data Fabric.

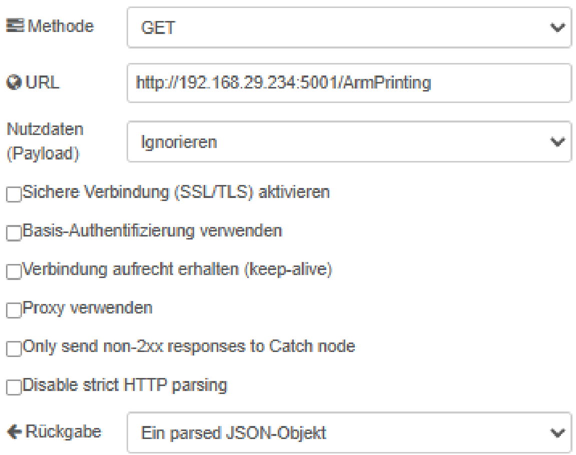


Figure 16. Pilot 1.1 http node for triggering the ArmPrinting endpoint in Node-RED


```json
{
    "energy_monitoring": {
        "available_data": ["power_current", "power_cumulated", "co2_footprint_current", "co2_footprint_cumulated"],
        "co2_footprint_cumulated": 29047.325,
        "co2_footprint_current": 0.015,
        "power_cumulated": 68.175,
        "power_current": 99.39
    },
    "environmental_data": {
        "atmospheric_pressure": 997,
        "available_data": ["temperature", "humidity", "vibration", "atmospheric_pressure"],
        "humidity": 59.6,
        "temperature": 22.2,
        "vibration": 5
    },
    "machine_status": {
        "machinename": "ArmPrinting",
        "orderid": "",
        "specific_machine_data": ["robot_data", "operating_mode"],
        "status": 1,
        "timestamp": 1755608163850
    },
    "specific_machine_data": [{
            "available_data": ["joint_1", "joint_2", "joint_3", "joint_4", "joint_5", "joint_6", "height", "velocity"],
            "height": 10.13,
            "id": "robot_data",
            "joint_1": 63.99,
            "joint_2": 98.53,
            "joint_3": 160.79,
            "joint_4": 125.71,
            "joint_5": 150.52,
            "joint_6": 18.82,
            "velocity": 4.06
        }, {
            "available_data": ["status_demomode"],
            "id": "operating_mode",
            "status_demomode": true
        }
    ]
```


Figure 17. Pilot 1.1 ArmPrinting Data Structure after triggering the asset endpoint

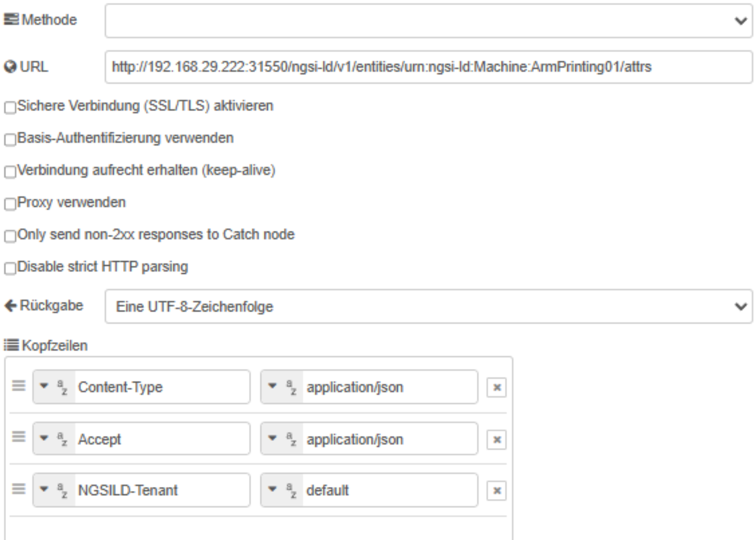


Figure 18. Pilot 1.1 http node for sending the ArmPrinting data to the Orion-LD ArmPrinting

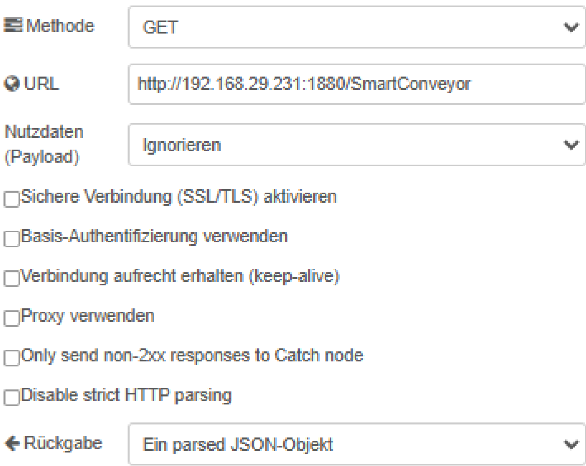


Figure 19. Pilot 1.1 http node for triggering the SmartConveyor endpoint in Node-RED


```
{
    "energy_monitoring": {
        "available_data": ["power_current", "power_cumulated", "co2_footprint_current", "co2_footprint_cumulated"],
        "co2_footprint_cumulated": 48359.36,
        "co2_footprint_current": 0.011,
        "power_cumulated": 14.438,
        "power_current": 19.58
    },
    "environmental_data": {
        "atmospheric_pressure": 1020,
        "available_data": ["temperature", "humidity", "vibration", "atmospheric_pressure"],
        "humidity": 39.4,
        "temperature": 18.1,
        "vibration": 4.984
    },
    "machine_status": {
        "machinename": "SmartConveyor",
        "orderid": "",
        "specific_machine_data": ["conveyor_data"],
        "status": 1,
        "timestamp": 1755608398592
    },
    "specific_machine_data": [{
            "available_data": ["conveyor_velocity"],
            "conveyor_velocity": 0,
            "id": "conveyor_data"
        }
    ]
}
```


Figure 20. Pilot 1.1 SmartConveyor Data Structure after triggering the asset endpoint

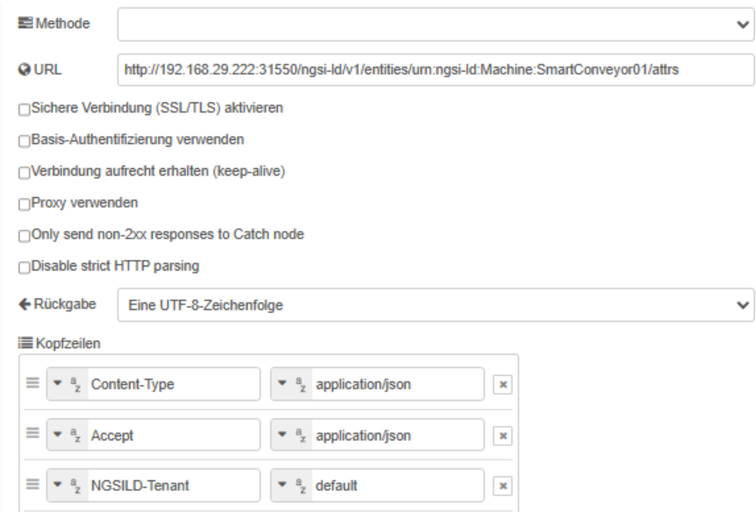


Figure 21. Pilot 1.1 http node for sending the SmartConveyor data to the Orion-LD SmartConveyor entity

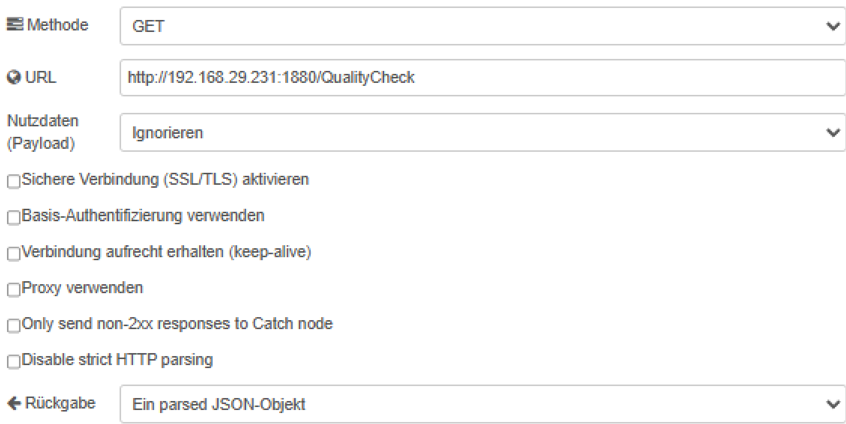


Figure 22. Pilot 1.1 http node for triggering the QualityCheck endpoint in Node-RED


```
{
    "energy_monitoring": {
        "available_data": ["power_current", "power_cumulated", "co2_footprint_current", "co2_footprint_cumulated"],
        "co2_footprint_cumulated": 24365.136,
        "co2_footprint_current": 0.014,
        "power_cumulated": 11.082,
        "power_current": 14.18
    },
    "environmental_data": {
        "atmospheric_pressure": 999,
        "available_data": ["temperature", "humidity", "vibration", "atmospheric_pressure"],
        "humidity": 31.8,
        "temperature": 25,
        "vibration": 5
    },
    "machine_status": {
        "machinename": "QualityCheck",
        "orderid": "",
        "specific_machine_data": [],
        "status": 1,
        "timestamp": 1755608670187
    }
}
```


Figure 23. Pilot 1.1 QualityCheck data structure after triggering the asset endpoint

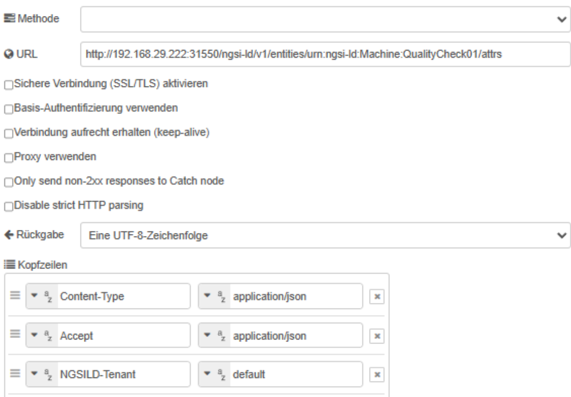


Figure 24. Pilot 1.1 http node for sending the QualityCheck data to the Orion-LD QualityCheck entity

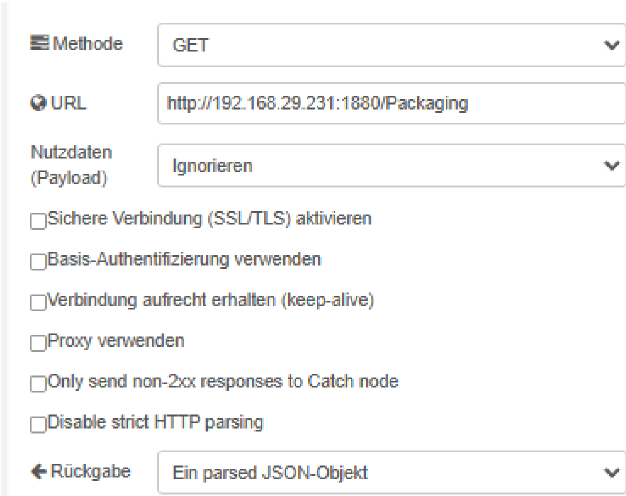


Figure 25. Pilot 1.1 http node for triggering the Packaging endpoint in Node-RED


```
    "energy_monitoring": {
        "available_data": ["power_current", "power_cumulated", "co2_footprint_current", "co2_footprint_cumulated"],
        "co2_footprint_cumulated": 9182.682,
        "co2_footprint_current": 0.011,
        "power_cumulated": 4.536,
        "power_current": 9.15
    },
    "environmental_data": {
        "atmospheric_pressure": 961,
        "available_data": ["temperature", "humidity", "vibration", "atmospheric_pressure"],
        "humidity": 61.5,
        "temperature": 17.8,
        "vibration": 4.995
    },
    "machine_status": {
        "machinename": "Packaging",
        "orderid": "",
        "specific_machine_data": [],
        "status": 1,
        "timestamp": 1750323643119
    }
}
```


Figure 26. Pilot 1.1 Pilot 1.1 Data Structure after triggering the asset endpoint

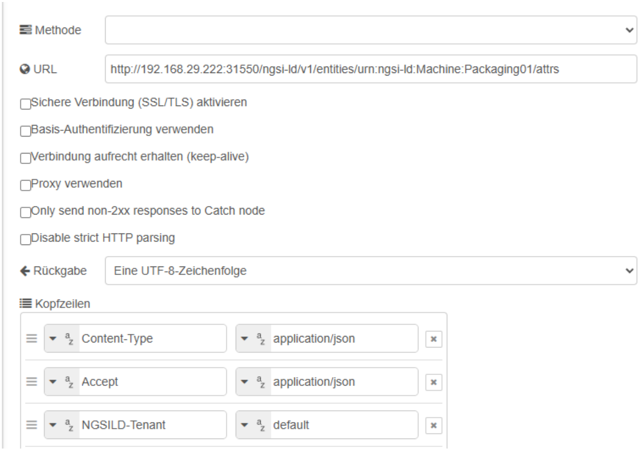


Figure 27. Pilot 1.1 http node for sending the Packaging data to the Orion-LD Packaging entity

The Orion-LD Context Broker manages the data model of each station in a different entity by storing the most recent values for their attributes. These values are accessed by the embedded-analytics-tool function through a direct http request to a REST API (e.g. http://192.168.29.222:31550/ngsi-ld/v1/entities/urn:ngsi-ld:Machine:Packaging01)

Pilot1 - Business Process 1 - Activity - 7 (P1-BP1-DA7): Software configuration / development

The software configuration and development for analysing and visualizing the station is done.

One Embedded Analytics Tool (EAT) function is developed and deployed fetching the data from Orion-LD and pushing it to the Prometheus Pushgateway, enabling the long-term storage in Prometheus.

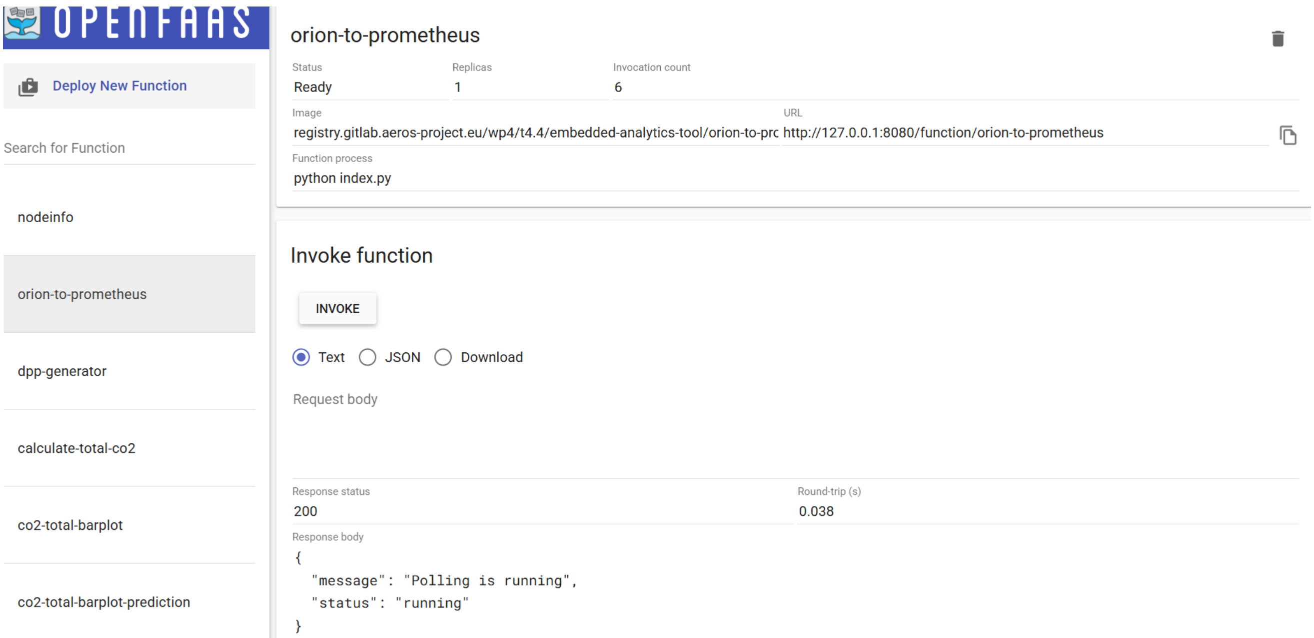


Figure 28. Pilot 1.1 orion-to-promethues EAT function visible inside the OpenFaaS UI. This function is responsible for the long-term storage of the data into PostgreSQL and Prometheus

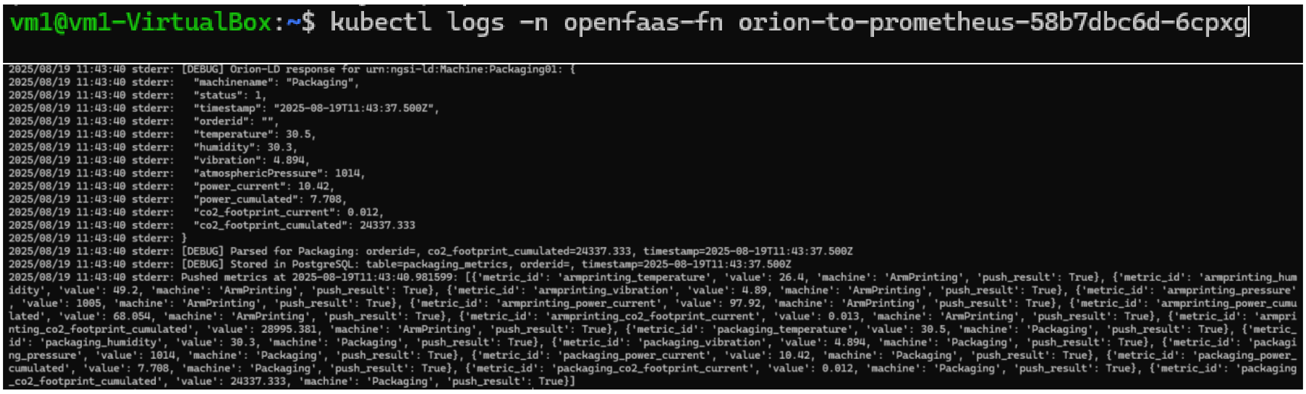


Figure 29. Pilot 1.1 Logs for orion-to-prometheus EAT function, showing the data retrieved by the Orion-LD Packaging entity and its storage in PostgreSQL and Prometheus

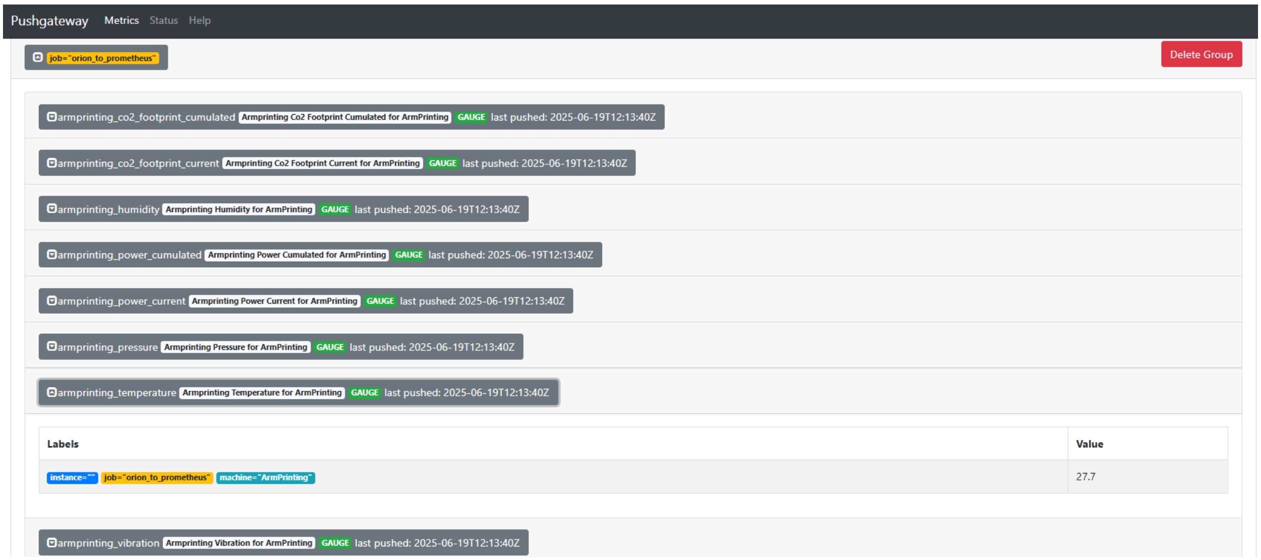


Figure 30. Pilot 1.1 Values Arm Printing station Pushed to the Prometheus push gateway

Figure 31. Pilot 1.1 Values (temperature) Arm Printing station stored in Prometheus displayed through the Prometheus UI

Another EAT function is developed and deployed monitor energy data and calculate the total CO2 footprint produced by each asset and drone order using the data stored in Prometheus and PostgreSQL. The evidence for this is shown in P1-BP1-IA14.

2.1.1.2.3. Integration activities

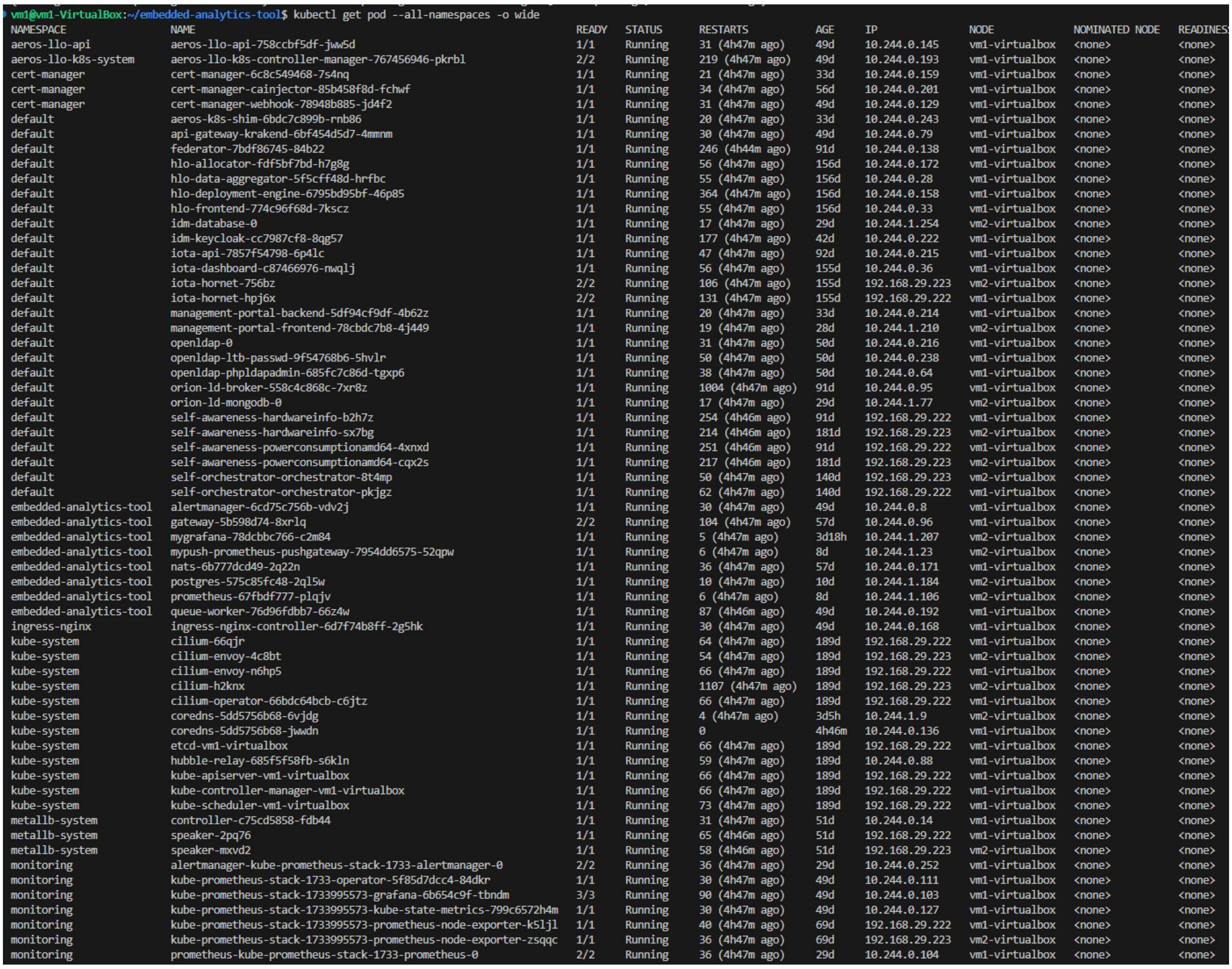
As mentioned in the introduction of D5.4, some stations initially identified to run pilot 1 scenario 1 were removed from the factory in March 2025. A few months were needed to build the new machines to be integrated into the "Lighthouse Factory Industry 4.0", which caused delay in integration activities and minor adaptations of the scenario. However, SIPBB ensured that it did not reduce the overall quality of the pilot and that enough data was collected to proceed with the validation activities as planned. Thus, this unexpected change in the logistics of the pilot was handled in a way that it did not disrupt the integration and validation activities. Instead, the team focused on software integration (aerOS Meta OS, own EAT functions, etc.), placing the validation for later (adjusted to the new timing of the project after amendment #2).

Pilot1 – Business Process 1 – Activity - 8 (P1-BP1-IA8): Identification of place of deployment in current architecture

aerOS components and related software are deployed within the Kubernetes cluster running with 2 Linux virtual machines. The virtual machines are managed with Oracle Virtual Box Manager running on a ProLiant MicroServer in the Swiss Smart Factory.

Figure 32. Pilot 1.1 ProLiant Micro Server connected to the SSF network

Figure 33. Pilot 1.1 Virtual Machines running on the ProLiant Micro Server



```
vm1@vm1-VirtualBox:~/embedded-analytics-tool$ kubectl get pod --all-namespaces -o wide
NAMESPACE                NAME                                                              READY   STATUS    RESTARTS          AGE     IP               NODE             NOMINATED NODE   READINES
aeros-llo-api            aeros-llo-api-758ccbf5df-jww5d                                    1/1     Running   31 (4h47m ago)    49d     10.244.0.145     vm1-virtualbox   <none>           <none>
aeros-llo-k8s-system     aeros-llo-k8s-controller-manager-767456946-pkrbl                  2/2     Running   219 (4h47m ago)   49d     10.244.0.193     vm1-virtualbox   <none>           <none>
cert-manager             cert-manager-6c8c549468-7s4nq                                     1/1     Running   21 (4h47m ago)    33d     10.244.0.159     vm1-virtualbox   <none>           <none>
cert-manager             cert-manager-cainjector-85b458f8d-fchwf                           1/1     Running   34 (4h47m ago)    56d     10.244.0.201     vm1-virtualbox   <none>           <none>
cert-manager             cert-manager-webhook-78948b885-jd4f2                              1/1     Running   31 (4h47m ago)    49d     10.244.0.129     vm1-virtualbox   <none>           <none>
default                  aeros-k8s-shim-6bdc7c899b-rnb86                                   1/1     Running   20 (4h47m ago)    33d     10.244.0.243     vm1-virtualbox   <none>           <none>
default                  api-gateway-krakend-6bf454d5d7-4mmnm                              1/1     Running   30 (4h47m ago)    49d     10.244.0.79      vm1-virtualbox   <none>           <none>
default                  federator-7bdf86745-84b22                                         1/1     Running   246 (4h44m ago)   91d     10.244.0.138     vm1-virtualbox   <none>           <none>
default                  hlo-allocator-fdf5bf7bd-h7g8g                                     1/1     Running   56 (4h47m ago)    156d    10.244.0.172     vm1-virtualbox   <none>           <none>
default                  hlo-data-aggregator-5f5cff48d-hrfbc                               1/1     Running   55 (4h47m ago)    156d    10.244.0.28      vm1-virtualbox   <none>           <none>
default                  hlo-deployment-engine-6795bd95bf-46p85                            1/1     Running   364 (4h47m ago)   156d    10.244.0.158     vm1-virtualbox   <none>           <none>
default                  hlo-frontend-774c96f68d-7kscz                                     1/1     Running   55 (4h47m ago)    156d    10.244.0.33      vm1-virtualbox   <none>           <none>
default                  idm-database-0                                                    1/1     Running   17 (4h47m ago)    29d     10.244.1.254     vm2-virtualbox   <none>           <none>
default                  idm-keycloak-cc7987cf8-8qg57                                      1/1     Running   177 (4h47m ago)   42d     10.244.0.222     vm1-virtualbox   <none>           <none>
default                  iota-api-7857f54798-6p4lc                                         1/1     Running   47 (4h47m ago)    92d     10.244.0.215     vm1-virtualbox   <none>           <none>
default                  iota-dashboard-c87466976-nwqlj                                    1/1     Running   56 (4h47m ago)    155d    10.244.0.36      vm1-virtualbox   <none>           <none>
default                  iota-hornet-756bz                                                 2/2     Running   106 (4h47m ago)   155d    192.168.29.223   vm2-virtualbox   <none>           <none>
default                  iota-hornet-hpj6x                                                 2/2     Running   131 (4h47m ago)   155d    192.168.29.222   vm1-virtualbox   <none>           <none>
default                  management-portal-backend-5df94cf9df-4b62z                        1/1     Running   20 (4h47m ago)    33d     10.244.0.214     vm1-virtualbox   <none>           <none>
default                  management-portal-frontend-78cbdc7b8-4j449                        1/1     Running   19 (4h47m ago)    28d     10.244.1.210     vm2-virtualbox   <none>           <none>
default                  openldap-0                                                        1/1     Running   31 (4h47m ago)    50d     10.244.0.216     vm1-virtualbox   <none>           <none>
default                  openldap-ltb-passwd-9f54768b6-5hvlr                               1/1     Running   50 (4h47m ago)    50d     10.244.0.238     vm1-virtualbox   <none>           <none>
default                  openldap-phpldapadmin-685fc7c86d-tgxp6                            1/1     Running   38 (4h47m ago)    50d     10.244.0.64      vm1-virtualbox   <none>           <none>
default                  orion-ld-broker-558c4c868c-7xr8z                                  1/1     Running   1004 (4h47m ago)  91d     10.244.0.95      vm1-virtualbox   <none>           <none>
default                  orion-ld-mongodb-0                                                1/1     Running   17 (4h47m ago)    29d     10.244.1.77      vm2-virtualbox   <none>           <none>
default                  self-awareness-hardwareinfo-b2h7z                                 1/1     Running   254 (4h46m ago)   91d     192.168.29.222   vm1-virtualbox   <none>           <none>
default                  self-awareness-hardwareinfo-sx7bg                                 1/1     Running   214 (4h46m ago)   181d    192.168.29.223   vm2-virtualbox   <none>           <none>
default                  self-awareness-powerconsumptionamd64-4xnxd                        1/1     Running   251 (4h46m ago)   91d     192.168.29.222   vm1-virtualbox   <none>           <none>
default                  self-awareness-powerconsumptionamd64-cqx2s                        1/1     Running   217 (4h46m ago)   181d    192.168.29.223   vm2-virtualbox   <none>           <none>
default                  self-orchestrator-orchestrator-8t4mp                              1/1     Running   50 (4h47m ago)    140d    192.168.29.223   vm2-virtualbox   <none>           <none>
default                  self-orchestrator-orchestrator-pkjgz                              1/1     Running   62 (4h47m ago)    140d    192.168.29.222   vm1-virtualbox   <none>           <none>
embedded-analytics-tool  alertmanager-6cd75c756b-vdv2j                                     1/1     Running   30 (4h47m ago)    49d     10.244.0.8       vm1-virtualbox   <none>           <none>
embedded-analytics-tool  gateway-5b598d74-8xrlq                                            2/2     Running   104 (4h47m ago)   57d     10.244.0.96      vm1-virtualbox   <none>           <none>
embedded-analytics-tool  mygrafana-78dcbbc766-c2m84                                        1/1     Running   5 (4h47m ago)     3d18h   10.244.1.207     vm2-virtualbox   <none>           <none>
embedded-analytics-tool  mypush-prometheus-pushgateway-7954dd6575-52qpw                    1/1     Running   6 (4h47m ago)     8d      10.244.1.23      vm2-virtualbox   <none>           <none>
embedded-analytics-tool  nats-6b777dcd49-2q22n                                             1/1     Running   36 (4h47m ago)    57d     10.244.0.171     vm1-virtualbox   <none>           <none>
embedded-analytics-tool  postgres-575c85fc48-2ql5w                                         1/1     Running   10 (4h47m ago)    10d     10.244.1.184     vm2-virtualbox   <none>           <none>
embedded-analytics-tool  prometheus-67fbdf777-plqjv                                        1/1     Running   6 (4h47m ago)     8d      10.244.1.106     vm2-virtualbox   <none>           <none>
embedded-analytics-tool  queue-worker-76d96fdbb7-66z4w                                     1/1     Running   87 (4h46m ago)    49d     10.244.0.192     vm1-virtualbox   <none>           <none>
ingress-nginx            ingress-nginx-controller-6d7f74b8ff-2g5hk                         1/1     Running   30 (4h47m ago)    49d     10.244.0.168     vm1-virtualbox   <none>           <none>
kube-system              cilium-66qjr                                                      1/1     Running   64 (4h47m ago)    189d    192.168.29.222   vm1-virtualbox   <none>           <none>
kube-system              cilium-envoy-4c8bt                                                1/1     Running   54 (4h47m ago)    189d    192.168.29.223   vm2-virtualbox   <none>           <none>
kube-system              cilium-envoy-n6hp5                                                1/1     Running   66 (4h47m ago)    189d    192.168.29.222   vm1-virtualbox   <none>           <none>
kube-system              cilium-h2knx                                                      1/1     Running   1107 (4h47m ago)  189d    192.168.29.223   vm2-virtualbox   <none>           <none>
kube-system              cilium-operator-66bdc64bcb-c6jtz                                  1/1     Running   66 (4h47m ago)    189d    192.168.29.222   vm1-virtualbox   <none>           <none>
kube-system              coredns-5dd5756b68-6vjdg                                          1/1     Running   4 (4h47m ago)     3d5h    10.244.1.9       vm2-virtualbox   <none>           <none>
kube-system              coredns-5dd5756b68-jwwdn                                          1/1     Running   0                 4h46m   10.244.0.136     vm1-virtualbox   <none>           <none>
kube-system              etcd-vm1-virtualbox                                               1/1     Running   66 (4h47m ago)    189d    192.168.29.222   vm1-virtualbox   <none>           <none>
kube-system              hubble-relay-685f5f58fb-s6kln                                     1/1     Running   59 (4h47m ago)    189d    10.244.0.88      vm1-virtualbox   <none>           <none>
kube-system              kube-apiserver-vm1-virtualbox                                     1/1     Running   66 (4h47m ago)    189d    192.168.29.222   vm1-virtualbox   <none>           <none>
kube-system              kube-controller-manager-vm1-virtualbox                            1/1     Running   66 (4h47m ago)    189d    192.168.29.222   vm1-virtualbox   <none>           <none>
kube-system              kube-scheduler-vm1-virtualbox                                     1/1     Running   73 (4h47m ago)    189d    192.168.29.222   vm1-virtualbox   <none>           <none>
metallb-system           controller-c75cd5858-fdb44                                        1/1     Running   31 (4h47m ago)    51d     10.244.0.14      vm1-virtualbox   <none>           <none>
metallb-system           speaker-2pq76                                                     1/1     Running   65 (4h46m ago)    51d     192.168.29.222   vm1-virtualbox   <none>           <none>
metallb-system           speaker-mxvd2                                                     1/1     Running   58 (4h46m ago)    51d     192.168.29.223   vm2-virtualbox   <none>           <none>
monitoring               alertmanager-kube-prometheus-stack-1733-alertmanager-0            2/2     Running   36 (4h47m ago)    29d     10.244.0.252     vm1-virtualbox   <none>           <none>
monitoring               kube-prometheus-stack-1733-operator-5f85d7dcc4-84dkr              1/1     Running   30 (4h47m ago)    49d     10.244.0.111     vm1-virtualbox   <none>           <none>
monitoring               kube-prometheus-stack-1733995573-grafana-6b654c9f-tbndm           3/3     Running   90 (4h47m ago)    49d     10.244.0.103     vm1-virtualbox   <none>           <none>
monitoring               kube-prometheus-stack-1733995573-kube-state-metrics-799c6572h4m   1/1     Running   30 (4h47m ago)    49d     10.244.0.127     vm1-virtualbox   <none>           <none>
monitoring               kube-prometheus-stack-1733995573-prometheus-node-exporter-k5ljl   1/1     Running   40 (4h47m ago)    69d     192.168.29.222   vm1-virtualbox   <none>           <none>
monitoring               kube-prometheus-stack-1733995573-prometheus-node-exporter-zsqqc   1/1     Running   36 (4h47m ago)    69d     192.168.29.223   vm2-virtualbox   <none>           <none>
monitoring               prometheus-kube-prometheus-stack-1733-prometheus-0                2/2     Running   36 (4h47m ago)    29d     10.244.0.104     vm1-virtualbox   <none>           <none>
```

Figure 34. Pilot 1.1 Kubernetes Kluster including pod for aerOS components, embedded-analytics-tool and related software running on VM1 control panel.

Pilot1 – Business Process 1 – Activity – 9 (P1-BP1-IA9): Requirements definition for deployment

The requirements for our deployments are as follows:

1- Running Kubernetes Cluster
2- EAT tool installed and related pods running in Kubernetes cluster such as:
    - Prometheus Push gateway
    - Prometheus
    - EAT Gateway
    - Grafana
    - EAT functions
        - **orion-to-prometheus**: Stores data to PostgreSQL and Prometheus
        - **calculate-total-co2**: Takes co2 data from Prometheus and calculates the total for each machine and drone
        - **dpp-generator**: Generates a digital product passport for each drone containing co2 footprint information
        - **co2-total-barplot-prediction**: Plots the co2 footprint for each machine and drone in Grafana and makes co2 footprint prediction for the next drone order.

Pilot1 – Business Process 1 – Activity - 10 (P1-BP1-IA10): Test energy data collection

Data collection involves triggering the machine endpoint which are exposed via http, forwarding the data to Orion-LD and storing it long-term in PostGreSQL and Prometheus

Orion-LD maintains the current state of the machine data and serves as a central hub for managing and distributing this information to the databases and analytic functions.

This pipeline has been successfully set up and tested. The Node-RED logs and the database queries shown below confirm the successful implementation of the energy data collection pipeline.


```
20.8.2025, 09:48:05   node: Fetched Raw ArmPrinting Data
msg.payload : Object
▾object
 ▾energy_monitoring: object
   ▾available_data: array[4]
      0: "power_current"
      1: "power_cumulated"
      2: "co2_footprint_current"
      3: "co2_footprint_cumulated"
    co2_footprint_cumulated: 29869.055
    co2_footprint_current: 0.02
    power_cumulated: 70.061
    power_current: 101.16
 ▾environmental_data: object
    atmospheric_pressure: 1016
   ▾available_data: array[4]
      0: "temperature"
      1: "humidity"
      2: "vibration"
      3: "atmospheric_pressure"
    humidity: 56.7
    temperature: 21.8
    vibration: 5
 ▾machine_status: object
    machinename: "ArmPrinting"
    orderid: ""
   ▸specific_machine_data: array[2]
    status: 1
    timestamp: 1755676085133
 ▾specific_machine_data: array[2]
   ▾0: object
     ▸available_data: array[8]
      height: 49.4
      id: "robot_data"
      joint_1: 85.85
      joint_2: 121.43
      joint_3: 51.43
      joint_4: 134.22
      joint_5: 149.08
      joint_6: 19.11
      velocity: 9.8
   ▾1: object
     ▸available_data: array[1]
      id: "operating_mode"
      status_demomode: false
```


Figure 35. Pilot 1.1 JSON Data fetched from the machine endpoint logged in Node-RED (First step in the pipeline)

19.8.2025, 16:03:52 node: Raw ArmPrinting Orion Response
msg.payload : Object
object
id: "urn:ngsi-ld:Machine:ArmPrinting01"
type: "Machine"
machinename: object
type: "Property"
value: "ArmPrinting"
status: object
type: "Property"
value: 1
timestamp: object
type: "Property"
value: "2025-08-19T14:03:51.821Z"
orderid: object
type: "Property"
value: ""
temperature: object
type: "Property"
value: 34.5
humidity: object
type: "Property"
value: 45.9
vibration: object
type: "Property"
value: 4.994
atmosphericPressure: object
type: "Property"
value: 990
power_current: object
type: "Property"
value: 99.6
power_cumulated: object
type: "Property"
value: 68.288
co2_footprint_current: object
type: "Property"
value: 0.014
co2_footprint_cumulated: object
type: "Property"
value: 29094.835

Figure 36. Pilot 1.1 Current state of ArmPrinting maintained by Orion-LD logged in Node-RED (Second step in the data collection pipeline)

Figure 37. Pilot 1.1 Long term storage of co2_footprint_current inside Prometheus DB

```
vm1@vm1-VirtualBox:~$ kubectl exec -n embedded-analytics-tool -it $(kubectl get pod -n embedded-analytics-tool -l app=postgres -o jsonpath='{.items[0].metadata.name}') -- psql -U eat_user -d eat_db -c "SELECT
*  FROM armprinting_metrics ORDER BY timestamp DESC LIMIT 10;"
        timestamp         | temperature | humidity | vibration | atmosphericpressure | power_current | power_cumulated | co2_footprint_current | co2_footprint_cumulated | orderid
--------------------------+-------------+----------+-----------+---------------------+---------------+-----------------+-----------------------+-------------------------+---------
 2025-08-19T14:05:50.263Z |        34.8 |     46.9 |     4.944 |                 993 |        100.09 |          68.291 |                 0.011 |                29096.23 |
 2025-08-19T14:05:41.182Z |        34.8 |     46.3 |         5 |                 991 |        100.62 |          68.291 |                 0.011 |               29096.104 |
 2025-08-19T14:05:35.078Z |        34.7 |       48 |         5 |                 989 |         98.91 |          68.291 |                 0.013 |               29096.018 |
 2025-08-19T14:05:25.980Z |        34.9 |     48.2 |     4.917 |                 991 |         99.73 |           68.29 |                 0.009 |               29095.908 |
 2025-08-19T14:05:16.903Z |        34.8 |     46.4 |     4.749 |                 991 |        100.96 |           68.29 |                 0.009 |                29095.83 |
 2025-08-19T14:05:07.823Z |        34.6 |     46.6 |     4.888 |                 991 |         99.88 |           68.29 |                 0.013 |               29095.727 |
 2025-08-19T14:05:01.700Z |        34.6 |     44.7 |     4.857 |                 990 |         99.97 |           68.29 |                 0.009 |                29095.66 |
 2025-08-19T14:04:52.530Z |        34.3 |       45 |     4.984 |                 991 |         99.08 |          68.289 |                 0.004 |               29095.604 |
 2025-08-19T14:04:43.379Z |        34.1 |     44.2 |         5 |                 989 |        100.74 |          68.289 |                 0.015 |               29095.535 |
 2025-08-19T14:04:34.285Z |          34 |     43.5 |         5 |                 990 |         99.87 |          68.289 |                 0.008 |               29095.398 |
```

Figure 38. Pilot 1.1 Long term storage of ArmPrinting Data inside PosgreSQL Database

Pilot1 – Business Process 1 – Activity - 11 (P1-BP1-IA11): Adjust equipment configuration according to first feedback

After the initial data collection tests, an evaluation was performed on whether any hardware or configuration changes were needed. The system feedback was positive, and all stations responded correctly to data queries. It was found that the initial configuration was sufficient – the network was stable, and each device provided data in the expected format. Thus, no major adjustments to the equipment setup were necessary at this stage. The Raspberry Pi and server handled the data rates without issue, and the factory Wi-Fi bandwidth proved adequate for the small-size messages we transmit. This activity was therefore concluded with the confirmation that the equipment and IT setup were correctly tuned, and we could proceed to more advanced integration steps.

Pilot1 – Business Process 1 – Activity - 12 (P1-BP1-IA12): aerOS Basic components

Concurrently, it was ensured that all basic aerOS components were up and running in SIPBB domain. This includes fundamental components like the High-Level Orchestrator (HLO) and Low-Level Orchestrator (LLO), network manager and security services, provided by the aerOS framework. Figure 39 shows that these services were successfully deployed in SIPBB domain. Having the basic components operational means our edge domain is recognized as part of the aerOS ecosystem and can be centrally monitored and managed.

```
vm1@vm1-VirtualBox:~$ kubectl get pods --all-namespaces
NAMESPACE                NAME                                                              READY   STATUS    RESTARTS        AGE
aeros-llo-api            aeros-llo-api-758ccbf5df-jww5d                                    1/1     Running   43 (42h ago)    111d
aeros-llo-k8s-system     aeros-llo-k8s-controller-manager-767456946-pkrbl                  2/2     Running   322 (41h ago)   111d
cert-manager             cert-manager-6c8c549468-7s4nq                                     1/1     Running   33 (42h ago)    95d
cert-manager             cert-manager-cainjector-85b458f8d-fchwf                           1/1     Running   46 (42h ago)    117d
cert-manager             cert-manager-webhook-78948b885-jd4f2                              1/1     Running   43 (42h ago)    111d
default                  aeros-k8s-shim-6bdc7c899b-rnb86                                   1/1     Running   32 (42h ago)    95d
default                  api-gateway-krakend-6bf454d5d7-4mmnm                              1/1     Running   42 (42h ago)    111d
default                  federator-7bdf86745-w2dz4                                         1/1     Running   0               41h
default                  hlo-allocator-fdf5bf7bd-8gcg6                                     1/1     Running   0               23m
default                  hlo-data-aggregator-5f5cff48d-hrfbc                               1/1     Running   67 (42h ago)    217d
default                  hlo-deployment-engine-6795bd95bf-46p85                            1/1     Running   376 (42h ago)   217d
default                  hlo-frontend-774c96f68d-7kscz                                     1/1     Running   67 (42h ago)    217d
default                  idm-database-0                                                    1/1     Running   3 (42h ago)     46h
default                  iota-api-7857f54798-6p4lc                                         1/1     Running   59 (42h ago)    153d
default                  iota-dashboard-c87466976-z5m49                                    1/1     Running   12 (42h ago)    47d
default                  iota-hornet-756bz                                                 2/2     Running   128 (42h ago)   216d
default                  iota-hornet-mng7v                                                 2/2     Running   20 (42h ago)    47d
default                  management-portal-backend-5df94cf9df-4b62z                        1/1     Running   32 (42h ago)    95d
default                  management-portal-frontend-78cbdc7b8-gjd8x                        1/1     Running   3 (42h ago)     8d
default                  openldap-0                                                        1/1     Running   9 (42h ago)     47d
default                  openldap-ltb-passwd-9f54768b6-chw64                               1/1     Running   4 (42h ago)     8d
default                  openldap-phpldapadmin-685fc7c86d-9x48v                            1/1     Running   3 (42h ago)     8d
default                  orion-ld-broker-558c4c868c-f9w92                                  1/1     Running   0               41h
default                  orion-ld-mongodb-0                                                1/1     Running   3 (42h ago)     46h
default                  self-awareness-hardwareinfo-mp6t7                                 1/1     Running   4 (41h ago)     41h
default                  self-awareness-hardwareinfo-sx7bg                                 1/1     Running   315 (41h ago)   242d
default                  self-awareness-powerconsumptionamd64-4xnxd                        1/1     Running   355 (42h ago)   153d
default                  self-awareness-powerconsumptionamd64-wp4zf                        1/1     Running   0               41h
default                  self-orchestrator-orchestrator-8t4mp                              1/1     Running   61 (42h ago)    202d
default                  self-orchestrator-orchestrator-pkjgz                              1/1     Running   75 (42h ago)    202d
embedded-analytics-tool  alertmanager-6cd75c756b-vdv2j                                     1/1     Running   42 (42h ago)    111d
embedded-analytics-tool  gateway-5b598d74-s7xml                                            2/2     Running   8 (42h ago)     8d
embedded-analytics-tool  mygrafana-78dcbbc766-hrb67                                        1/1     Running   0               41h
embedded-analytics-tool  mypush-prometheus-pushgateway-7954dd6575-mmm6z                    1/1     Running   3 (42h ago)     8d
embedded-analytics-tool  nats-6b777dcd49-2q22n                                             1/1     Running   48 (42h ago)    119d
embedded-analytics-tool  postgres-575c85fc48-lfvzc                                         1/1     Running   3 (42h ago)     8d
embedded-analytics-tool  prometheus-67fbdf777-dp9w5                                        1/1     Running   3 (42h ago)     8d
embedded-analytics-tool  queue-worker-76d96fdbb7-gbtvs                                     1/1     Running   13 (42h ago)    8d
ingress-nginx            ingress-nginx-controller-6d7f74b8ff-2g5hk                         1/1     Running   42 (42h ago)    111d
kube-system              cilium-66qjr                                                      1/1     Running   77 (42h ago)    250d
kube-system              cilium-envoy-4c8bt                                                1/1     Running   65 (42h ago)    250d
kube-system              cilium-envoy-n6hp5                                                1/1     Running   79 (42h ago)    250d
kube-system              cilium-h2knx                                                      1/1     Running   1118 (42h ago)  250d
kube-system              cilium-operator-66bdc64bcb-c6jtz                                  1/1     Running   83 (42h ago)    250d
kube-system              coredns-5dd5756b68-4qcpn                                          1/1     Running   3 (42h ago)     46h
kube-system              coredns-5dd5756b68-mth7l                                          1/1     Running   0               42h
kube-system              etcd-vm1-virtualbox                                               1/1     Running   80 (42h ago)    250d
kube-system              hubble-relay-685f5f58fb-s6kln                                     1/1     Running   71 (42h ago)    250d
kube-system              kube-apiserver-vm1-virtualbox                                     1/1     Running   80 (42h ago)    250d
kube-system              kube-controller-manager-vm1-virtualbox                            1/1     Running   84 (42h ago)    250d
kube-system              kube-scheduler-vm1-virtualbox                                     1/1     Running   91 (42h ago)    250d
metallb-system           controller-c75cd5858-fdb44                                        1/1     Running   43 (42h ago)    112d
metallb-system           speaker-2pq76                                                     1/1     Running   90 (42h ago)    112d
metallb-system           speaker-mxvd2                                                     1/1     Running   81 (42h ago)    112d
monitoring               alertmanager-kube-prometheus-stack-1733-alertmanager-0            2/2     Running   60 (42h ago)    91d
monitoring               kube-prometheus-stack-1733-operator-5f85d7dcc4-84dkr              1/1     Running   44 (42h ago)    111d
monitoring               kube-prometheus-stack-1733995573-grafana-6b654c9f-r444w           3/3     Running   9 (42h ago)     8d
monitoring               kube-prometheus-stack-1733995573-kube-state-metrics-799c6572h4m   1/1     Running   42 (42h ago)    111d
monitoring               kube-prometheus-stack-1733995573-prometheus-node-exporter-k5ljl   1/1     Running   53 (42h ago)    131d
monitoring               kube-prometheus-stack-1733995573-prometheus-node-exporter-zsqqc   1/1     Running   48 (42h ago)    131d
monitoring               prometheus-kube-prometheus-stack-1733-prometheus-0                2/2     Running   6 (42h ago)     46h
openebs                  openebs-localpv-provisioner-66ddc8887c-ckslk                      1/1     Running   73 (42h ago)    250d
openebs                  openebs-ndm-87cst                                                 1/1     Running   131 (42h ago)   250d
openebs                  openebs-ndm-operator-64c5dbd564-ngvlz                             1/1     Running   71 (42h ago)    250d
openebs                  openebs-ndm-qgn2l                                                 1/1     Running   151 (42h ago)   250d
openfaas-fn              calculate-total-co2-79d9dfbcb4-vgpz7                              1/1     Running   80 (41h ago)    40d
openfaas-fn              co2-total-barplot-5dc758ddb8-54zdh                                1/1     Running   0               40h
openfaas-fn              co2-total-barplot-prediction-58dbcfb857-qhm62                     1/1     Running   0               40h
openfaas-fn              dpp-generator-6cfdcdd9bb-vhwd8                                    1/1     Running   0               40h
openfaas-fn              nodeinfo-77c6756589-nxhns                                         1/1     Running   60 (42h ago)    162d
openfaas-fn              orion-to-prometheus-58b7dbc6d-6cpxg                               1/1     Running   0               41h
redpanda                 redpanda-0                                                        2/2     Running   6 (42h ago)     46h
redpanda                 redpanda-console-7495f86cd9-ll5k6                                 1/1     Running   177 (42h ago)   153d
```

Figure 39. Pilot 1.1 aerOS Basic components installed.

Pilot1 – Business Process 1 – Activity - 13 (P1-BP1-IA13): aerOS non-basic components

The non-basic components required for Pilot 1.1 were installed and are listed in the image below. These components are part of the embedded-analytics-tool necessary for the long-term storage of the data, function deployment and dashboard visualization.

```
vm1@vm1-VirtualBox:~/embedded-analytics-tool$ kubectl get pod -n embedded-analytics-tool
$NAME                                             READY   STATUS    RESTARTS          AGE
alertmanager-6cd75c756b-vdv2j                     1/1     Running   30 (4h55m ago)    49d
gateway-5b598d74-8xrlq                            2/2     Running   104 (4h55m ago)   57d
mygrafana-78dcbbc766-c2m84                        1/1     Running   5 (4h55m ago)     3d18h
mypush-prometheus-pushgateway-7954dd6575-52qpw    1/1     Running   6 (4h55m ago)     8d
nats-6b777dcd49-2q22n                             1/1     Running   36 (4h55m ago)    57d
postgres-575c85fc48-2ql5w                         1/1     Running   10 (4h55m ago)    10d
prometheus-67fbdf777-plqjv                        1/1     Running   6 (4h55m ago)     8d
queue-worker-76d96fdbb7-66z4w                     1/1     Running   87 (4h53m ago)    49d
```

Figure 40. Pilot 1.1. aerOS non-basic components

Some non-basic components of aerOS were casted aside, since they did not align with the goals of the demonstrator, thus they were not essential for achieving the objectives of the pilot phase. In addition, some of these require further evaluation to confirm compatibility and meaningful integration or rely on prerequisites that are not yet in place, and will be documented in deliverable D5.6.

Pilot1 – Business Process 1 – Activity - 14 (P1-BP1-IA14): Dashboard creation with energy data

Two dashboards were created. The first shows the CO_2 footprint of the last six drone orders in a stacked bar plot. Each bar represents the total CO_2 emissions for a specific drone order, while the different colored segments indicate the contribution of each individual machine to the total.

The second dashboard displays the real-time energy consumption and CO_2 emissions of all machines. The values in this dashboard appear relatively low, as the machines were in standby mode and not processing a drone order at the time the screenshot was taken.

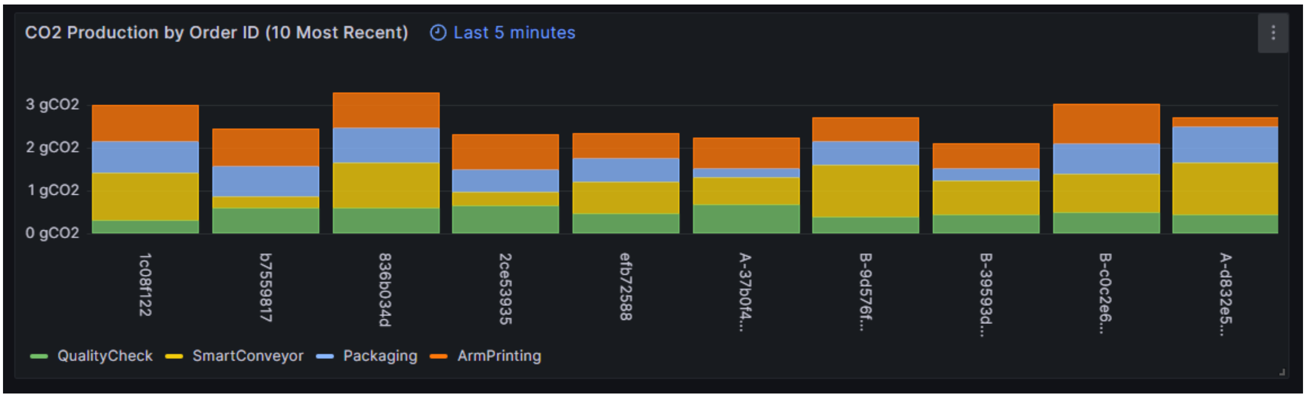


*Figure 41. **Pilot 1.1 Dashboard Displaying the carbon footprint for the last 10 drone orders.***

In the figure above, each bar chart represents the total carbon footprint for a drone order, with the different colored segments showing the carbon contribution of each individual machine

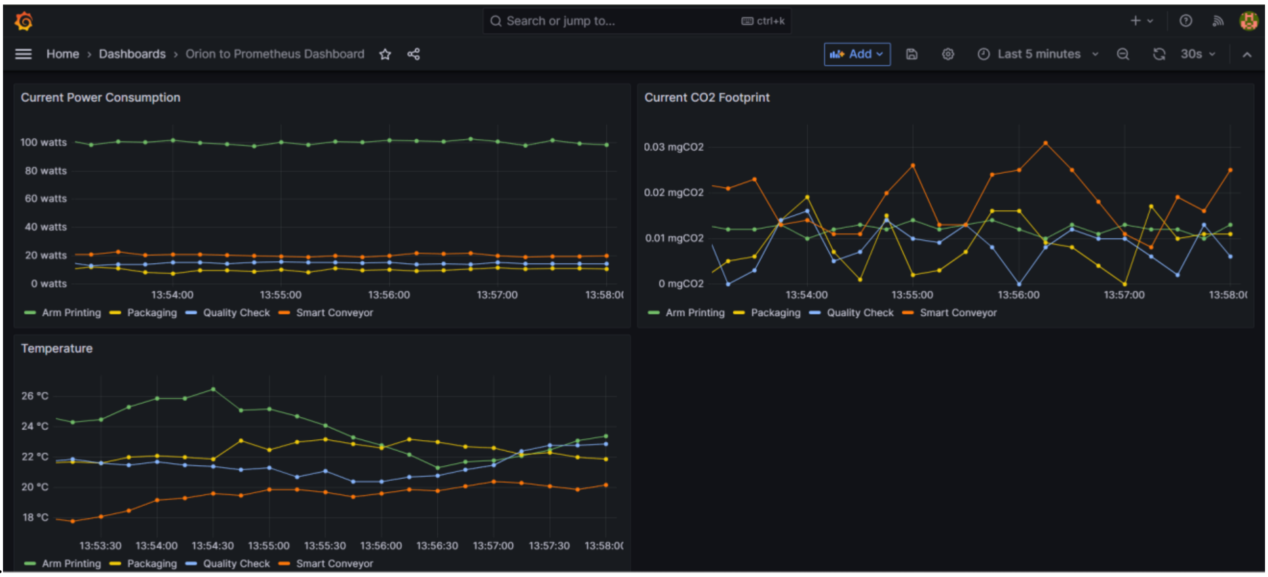


Figure 42. Pilot 1.1 Dashboard showing the real-time energy consumption of the assets in stand by modus.

Pilot1 – Business Process 1 – Activity - 15 (P1-BP1-IA15): Integration of data analysis service for reports and statistics creation

The DeepSeek-R1 1.5B LLM model for data analysis and report generation was tested within the aerOS deployment in Pilot 1.1, integrated as a cloud-native service (materialised as a Kubernetes pod) in the continuum of the use case. The goal of the reporting function is to support production optimization and the reduction of the overall CO_2 footprint.

The model takes production-related input data, such as machine-level energy consumption and process parameters, and generates reports that highlight patterns of energy use and their impact on emissions. These reports are designed to provide actionable insights, for example identifying which stations or drone types contribute most to CO_2 output and suggesting optimization strategies such as keeping high start-up-cost machines continuously running while only activating others when needed.

However, the available hardware resources proved insufficient to achieve satisfactory performance with a locally deployed model, especially since much of the computing capacity was already occupied by other software. For practical reasons, it was therefore decided to adapt the approach and integrate cloud-based LLMs via API instead.

As a result of this change in direction, the validation results of this task will be reported in deliverable D5.6 at the end of the project (M38).

Pilot1 – Business Process 1 – Activity - 16 (P1-BP1-IA16): Creation of a future prediction of the CO_2 footprint for each product

A predictive analytics function is used to forecast the CO_2 footprint of the next drone order.

Initially, an ARIMA model was tested; however, since the time series did not reveal substantial patterns that would benefit from a more complex model, a Moving Average approach was chosen. The function uses CO_2-per-product data from the last *N* orders (the next two Figures, for example, illustrate the footprints of the four most recent orders) and extrapolates a prediction for the next order by calculating the mean. This prediction is automatically updated as soon as new product data becomes available, following the principle of parsimony: always prefer the simplest adequate model.

The aerOS components provide the necessary data pipeline infrastructure, ensuring that the prediction function receives timely and consistent input data. In the Pilot 1.1, the experimentation has been tackled with two different drone types, for which separate CO_2 footprint predictions are generated. The Moving Average model uses the historical data of each specific drone type to forecast the footprint of the next order of that type.

The following two Figures show the carbon footprint of the last 5 orders of Type A and B drones and the resulting forecast for the next Type A and B drone, respectively. The different colours indicate how much each machine contributed to the carbon footprint of the order. The bar plot on the far right shows the prediction of the next order based on the moving average.

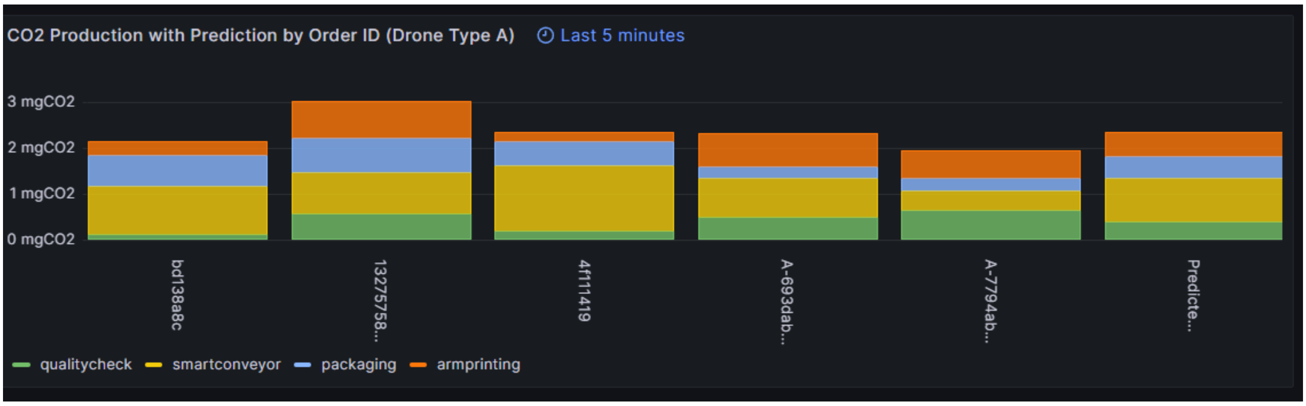


Figure 43. Pilot 1.1 Dashboard for Type A drones

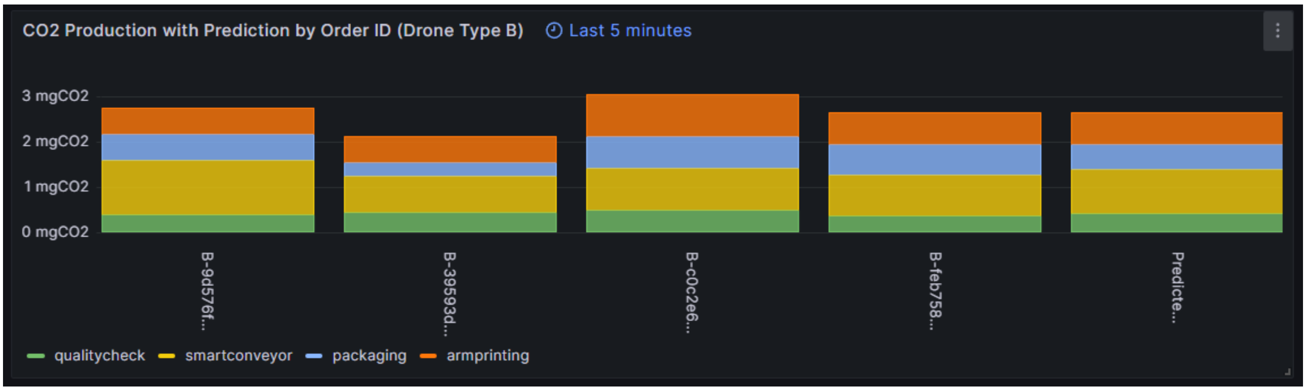


Figure 44. Pilot 1.1 Dashboard for Type B drones

Pilot1 – Business Process 1 – Activity - 17 (P1-BP1-IA17): Experimenting with Gaia-X services

Experimenting with the Gaia-X Wizard helped pilot team to understand and work with Verifiable Credentials (VCs) and Verifiable Presentations (VPs) which are key parts of the Gaia-X ecosystem. They enable **secure, decentralized trust and compliance verification** across different organizations.

Experimentation with the Gaia-X Playground enabled to create and sign a data resource example, as shown in the Figures below, derived from the Gaia-X playground templates:

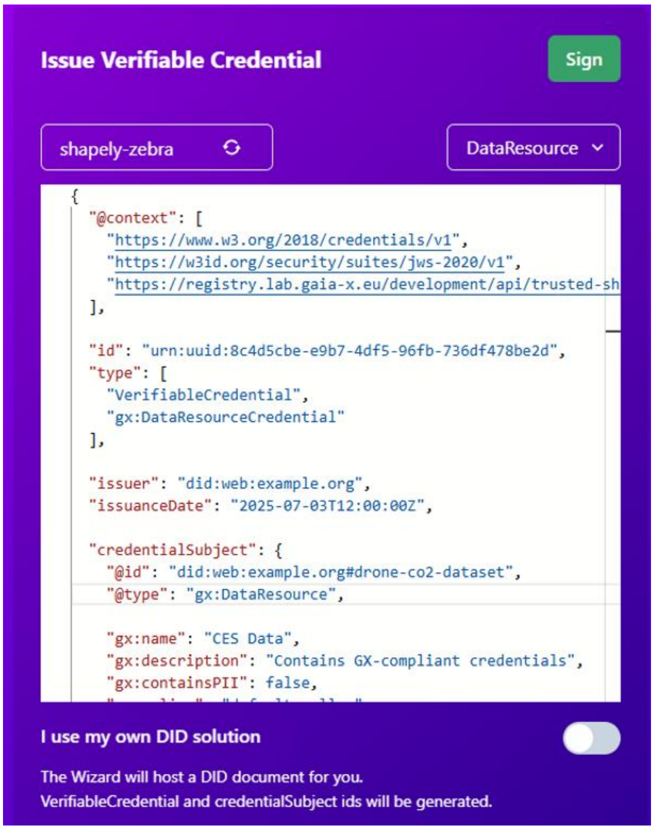


Figure 45. Pilot 1.1 Configured data resource as Gaia-X format

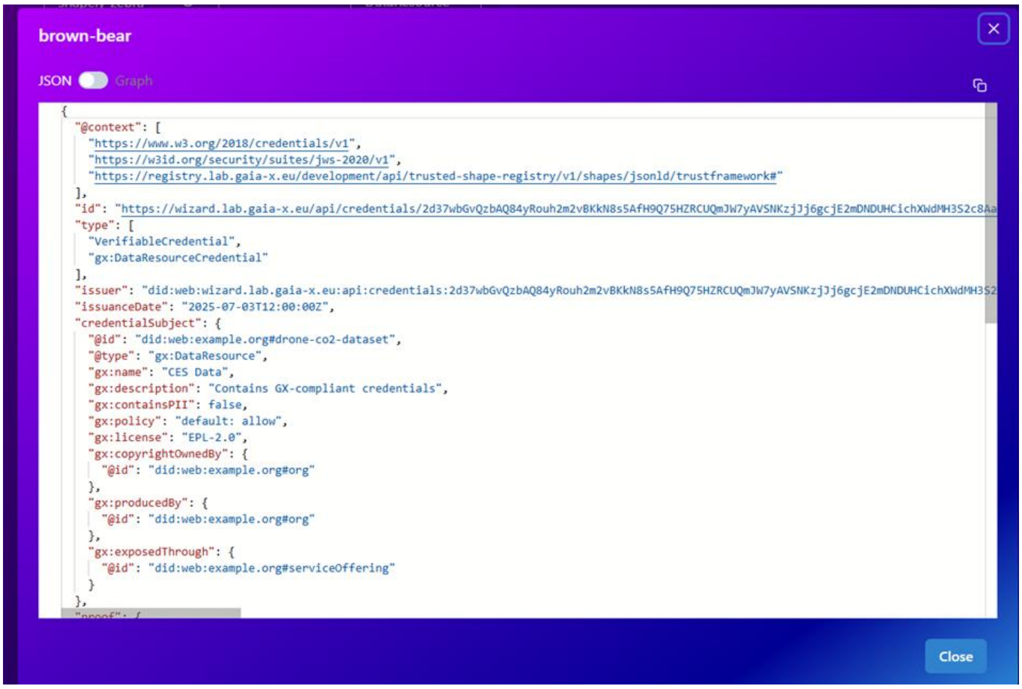


Figure 46. Pilot 1.1 Successfully Gaia-X signed document

A complete integration of Gaia-X within Pilot 1.1 was foreseen as a potential future working line, but not within the scope of the pilot in aerOS. This was decided due to the nature of the pilot's objective, which has been to showcase efficient, data-driven orchestration within the IoT edge-cloud continuum rather than validating compliance frameworks. Installing and maintaining the full Gaia-X stack, including connectors, trust frameworks, and governance services, would have added significant complexity and resource demands without increasing the demonstrative value for the industrial use cases addressed. Therefore, the Wizard-based exploration provided a lightweight yet effective means to familiarize with Gaia-X compliance processes, preserving focus on pilot execution and industrial relevance.

Pilot1 – Business Process 1 – Activity - 18 (P1-BP1-IA18): Digital Product Passport implementation

The primary achievement of scenario 1 is the successful implementation of a Digital Product Passport (DPP), specifically tailored to its drone production line, the "Lighthouse Factory Project Industry 4.0". After a careful evaluation of the parameters and KPIs considered, this DPP comprehensively encapsulates and includes all relevant assessed information and metrics through its lifecycle, from the importation of raw materials and parts to the completion of the drone. **This innovative tool establishes a comprehensive and structured system for documenting, analysing and evaluating key aspects of the production process. As such, it improves transparency, traceability, process efficiency and overall sustainability**. The DPP includes the following core functionalities:

- Comprehensive Documentation: It meticulously records all stages of the drone's upstream and manufacturing process, for the supplied components to those manufactured at SSF, providing a transparent overview from inception to completion. Each component is distinctly identified and accompanied by relevant information, including technical specifications and datasheets when available.
- Key Performance Indicators (KPIs): This DPP tracks critical Circularity, Sustainability, and Manufacturing KPIs across the production and lifecycle stages. Examples include metrics such as Cycle Time, Overall Operation Effectiveness (OOE), and the CO_2 footprint of individual drone components, offering valuable insights for process efficiency and further optimization.
- User Engagement: The DPP generates a unique QR code for each drone. This QR code provides users with access to detailed information regarding the product, while allowing them to document the drone's usage, further enhancing traceability and user involvement. This can aid in further product analysis of the downstream process, which culminates in the efficient remanufacturing or recycling of the drone.

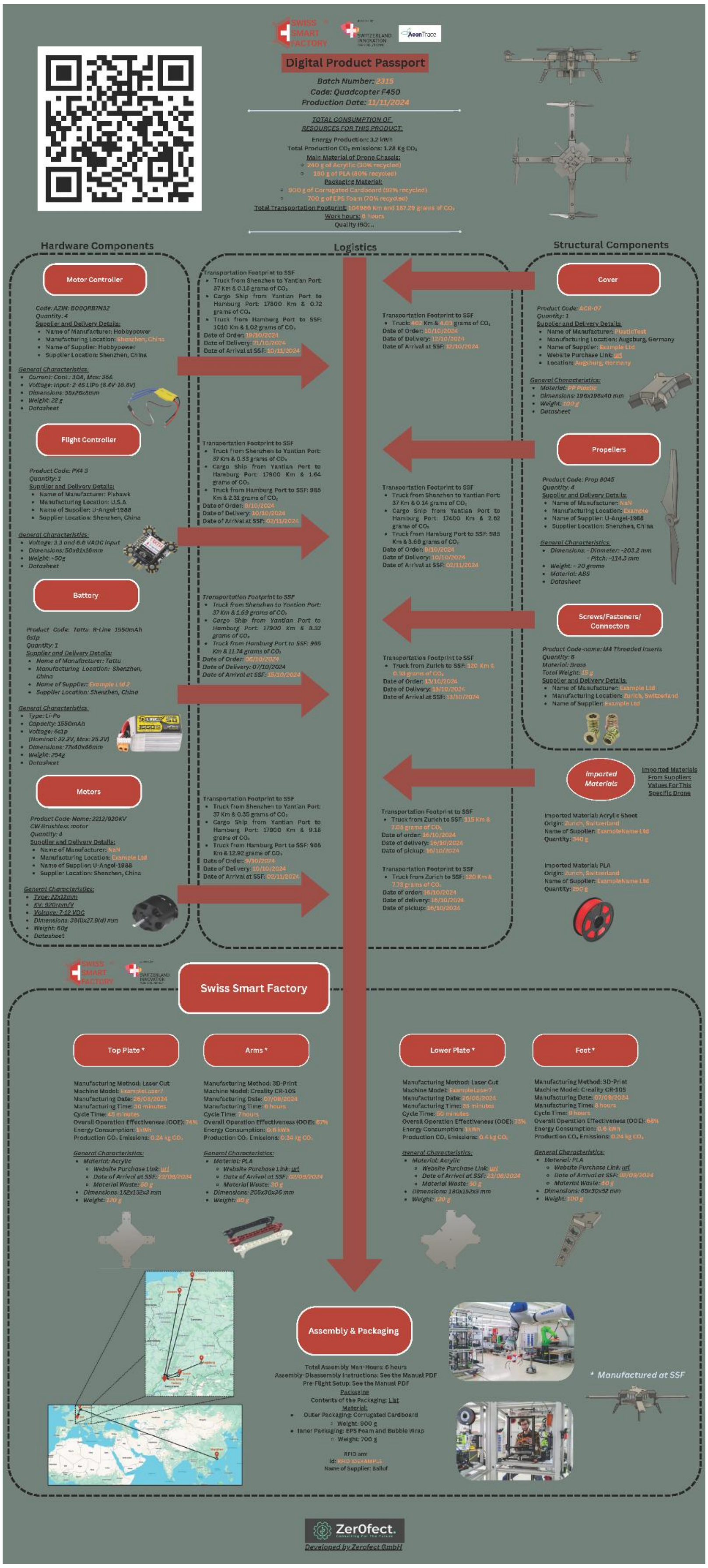


Figure 47. Pilot 1.1 Digital Product Passport implementation

Process Breakdown and **Key Achievements**:

1. **Component Classification**: The initial step was to categorize the components/parts in the two categories of supplied components and in-house components, as well as in two sub-categories of structural and hardware components, setting up the base to select and create the necessary data as well as the visual format of the DPP.
2. **Supplied Components**: After cataloguing and identifying all components/parts used in the drone's production, they were associated with their respective suppliers. This presented challenges due to the nature of our production setup, where we frequently source components from resellers rather than directly from manufacturers. Once the suppliers were identified, we successfully retrieved detailed technical information for each component as well as each respective manufacturer or supplier location. This step formed the foundation for establishing a robust database of supplied components.
3. **Logistics Analysis**: Using the geographic origins of each component, along with factors such as distance, transportation method and weight, we calculated the transport footprint associated with their delivery. This analysis provides an estimate of the environmental impact of the logistics process, contributing to a holistic understanding of the drone's overall carbon footprint. An Illustrative Transportation Map depicting the origin of imported parts and their route to SSF was created to provide a clearer perspective on the supply chain and logistics involved.
4. **In-House Components**: For components manufactured internally, we calculated data on energy consumption, CO_2 emissions, and other production-related metrics by taking into consideration the specific manufacturing method, the average energy consumption of the machines used, and the total manufacturing time. This detailed data compilation enables precise measurement of the environmental and resource impact of in-house production processes.
5. **Assembly and Packaging**: Information on potential packaging materials, weights, and related logistics was integrated into the DPP. This ensures that the environmental impact of the packaging process is accounted for, adding another layer of detail to the drone's lifecycle assessment.
6. **Total Footprint Calculation**: By combining data from supplied components, logistics, in-house manufacturing, and packaging, we calculated the total material and energy consumption for each drone. The resulting comprehensive footprint analysis serves as a benchmark for assessing and improving the sustainability of our production processes.

2.1.2. Automotive Smart Factory Zero Defect Manufacturing

The idea behind this pilot builds upon Innovalia's previous work in Metrology 4.0 and the need for a more flexible and resilient quality control processes. The first deployment—and Proof of Concept—was performed at NASERTIC digital premises (cloud-alike), where the Entry Domain services (identity management and access control) were initially hosted. This allowed the validation of secure authentication and standardized OPC-UA connectivity to Innovalia´s infrastructure.

Afterwards, and aligning with the exploitation approach of the main stakeholder, the pilot environment was fully migrated to Innovalia´s premises enabling the integration of aerOS components across multiple domains. At the end of this phase, complete data flows were established between the M3 Software, the Human Machine Interface (HMI), Edge devices hosting the measurement equipment (cameras and sensors)

The final stage is the real-world production metrology deployment at the AIC (Automotive Intelligence Center). Here, aerOS operated across a live industrial setup, connecting multiple CMMs, executing measurement sequences locally at the edge, assembling and synchronizing measurement and sensor data into Digital Twins, and enabling remote operation and reconfiguration. This not only proved the scalability of the architecture but also demonstrated tangible process improvements in responsiveness, resilience, and readiness for zero-defect manufacturing workflows. The following image shows the lab during the aerOS quality control testing.

Figure 48. AIC Metrology lab (Pilot 1.2)

The final deployment of Pilot 1.2 followed a structured path, beginning with infrastructure setup and progressing through development, integration, and validation phases.

Initial activities focused on preparing the metrology environment, including the installation of sensors and cameras on the CMMs and configuring the necessary infrastructure, already reported in detail within D5.3.

Development efforts enabled standardized communication with metrology equipment using OPC-UA, through the deployment of a dockerized RobotLink Open Platform Communication (RLOPC) service. This allowed real-time access to machine parameters and integration with interfaces such as M3 and the HMI.

Integration involved deploying a custom set of aerOS components across different domains, including authentication services, orchestration tools, and self-management modules. These enabled secure, modular, and coordinated operations across the metrology continuum.

Validation activities confirmed the successful remote configuration and operation of CMMs, the creation of Digital Twins for monitoring and allow potential deviations to be detected earlier, and the dynamic execution of measurement services. A Data Assembler was used to combine sensor and machine data, supporting efficient and semi-automated metrology workflows, merging inputs into coherent datasets (e.g. cloud or points from optical scanners, current state of CMM) and supporting the continuous update of the Digital twins

The experiment was designed as well **to demonstrate how aerOS can shift intelligence and control from the Cloud to the Edge, reducing latency, saving energy, and improving resilience and productivity**.

In regards of the switch of data processing from the Cloud to the Edge, the table below illustrates the evolution brought by the integration of the aerOS MetaOS. In the initial architecture, most data acquisition, processing, and decision making were performed centrally in the cloud, with edge devices playing a minimal role limited to data collection. After the implementation of aerOS, it was inverted. Most of the data processing is now executed at the edge level, orchestrated by the intelligence brought by the Meta OS, and still keeping the requirements of the industrial-like environment setup in AIC.

Key components—including the dockerized RLOPC services and Self* services—were deployed on edge nodes close to the measurement devices.

This enabled real-time configuration, operation, and monitoring of Coordinate Measuring Machines (CMMs), all within a secure, modular, and scalable framework. The new setup allows for increased autonomy of the edge, reduction in latency, and continuity of operations even during partial network disconnections.

Table 2. Pilot 1.2 Edge vs Cloud comparison before and after aerOS.

	Cloud	Edge
Before aerOS	- M3 Execution and orchestration logic executed remotely - Low responsiveness and high dependency on network stability Security and identification managed centrally	- Edge devices (e.g CMMs) with limited intelligence - No real-time control, only supervision No autonomous recovery or distributed intelligence
After aerOS	- Cloud layer used mainly for high-level analytics Acts as backup and coordination layer, not the primary executor	- Local execution of M3 services and CMM control via OPC-UA - Real-time decisions made locally through Self*modules - Faster configuration & setup and resilience thanks to Self-Healing and deployment services - More secure communications and identity management (Authentication control access to the OS and other services deployed) Deployment of services from any VM

2.1.2.1. Technical schema

This section describes the final architecture and technological scope of the pilot demonstrator, highlighting the different domains, infrastructure elements, and their roles in the metrology workflow.

The architecture is organized into 3 main domains, forming the metrology continuum:

1. **Entry Domain (Domain 1):** This domain handles user authentication and acts as the entry point to the system. It includes services such as Keycloak and OpenLDAP for identity management and was

initially hosted at NASERTIC premises during early testing phases. As the project advanced into a scaling & optimization phase, the Entry Domain was migrated to Innovalia servers, becoming the central gateway for system access.

2. **Innovalia Metrology Domain (Domain 2):** This domain includes the Infrastructure Elements (IEs) associated with the Coordinate Measuring Machines (CMMs). The main IE handles authentication and data exchange with Domain 1, while additional IEs—designated as IE Edge 2 and IE Edge 3—represent individual CMMs. These machines communicate using the OPC-UA Robot Link Server, enabling real-time access to their attributes and control methods. The CMMs operate without direct HLO installations and are instead linked to the edge infrastructure of Domain 2.
3. **M3 Software Domain (Domain 3):** This domain hosts the M3 metrology software, used by metrologists for monitoring and controlling CMM operations. It is connected to the Entry Domain through an OPC-UA Client, allowing seamless communication across domains. This setup enables remote access to the machines and real-time visualization of metrology tasks.

The following ontology diagram illustrates the relationship between these domains and their respective infrastructure elements, depicting a layered and distributed system aligned with aerOS MetOS principles.

Through its distributed architecture and with aerOS MetaOS support, the metrology process becomes both **predictive and adaptive**, rather than purely reactive. By ensuring data collection from multiple CMMs, sensors, and measurement workflows is continuously available and acted upon at the edge, the system can adjust operations in near real time to prevent potential quality issues. This capability reduces the likelihood of defects propagating through the production chain, thereby moving the manufacturing process measurably closer to a true zero-defect paradigm.

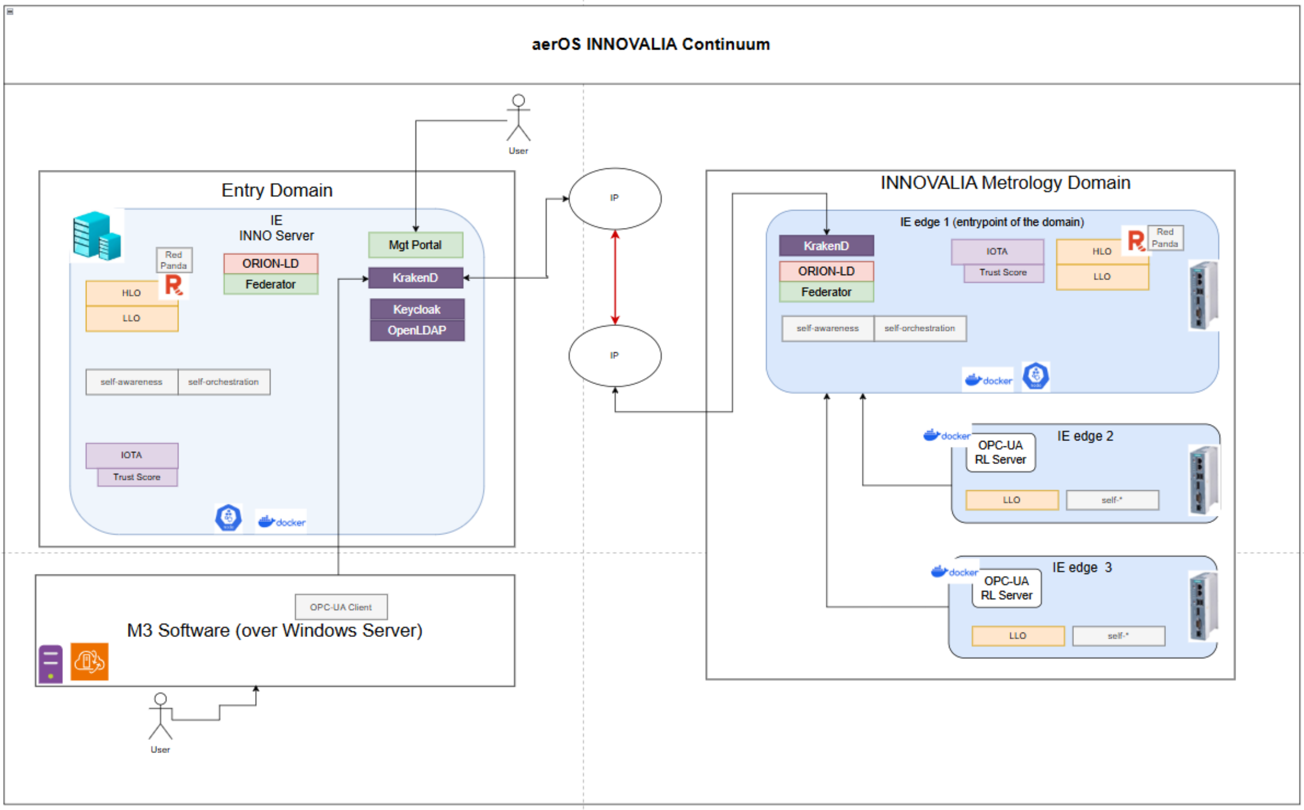


Figure 49. Pilot 1.2 Topology diagram – Technical Schema

2.1.2.2. Report of activities

2.1.2.2.1. Setup and procurement activities

Prior to the following report of work done, there are several activities successfully executed, especially those related with Setup and Procurement activities, and initial phase of development. For instance, the installation of control cameras and pressure sensors on the CMM, or the dimensioning of the NASERTIC VM and infrastructure. These activities refer to P1-BP2-SA1, P1-BP2-SA2, P1-BP2-SA3 and P1-BP2-SA4.

2.1.2.2.2. Development activities

Pilot 1 – Business Process 2 –Activity 5 (P1-BP2-DA5) – Enable Software-based control services and IoT hub for collection and brokering for instrumentation information:

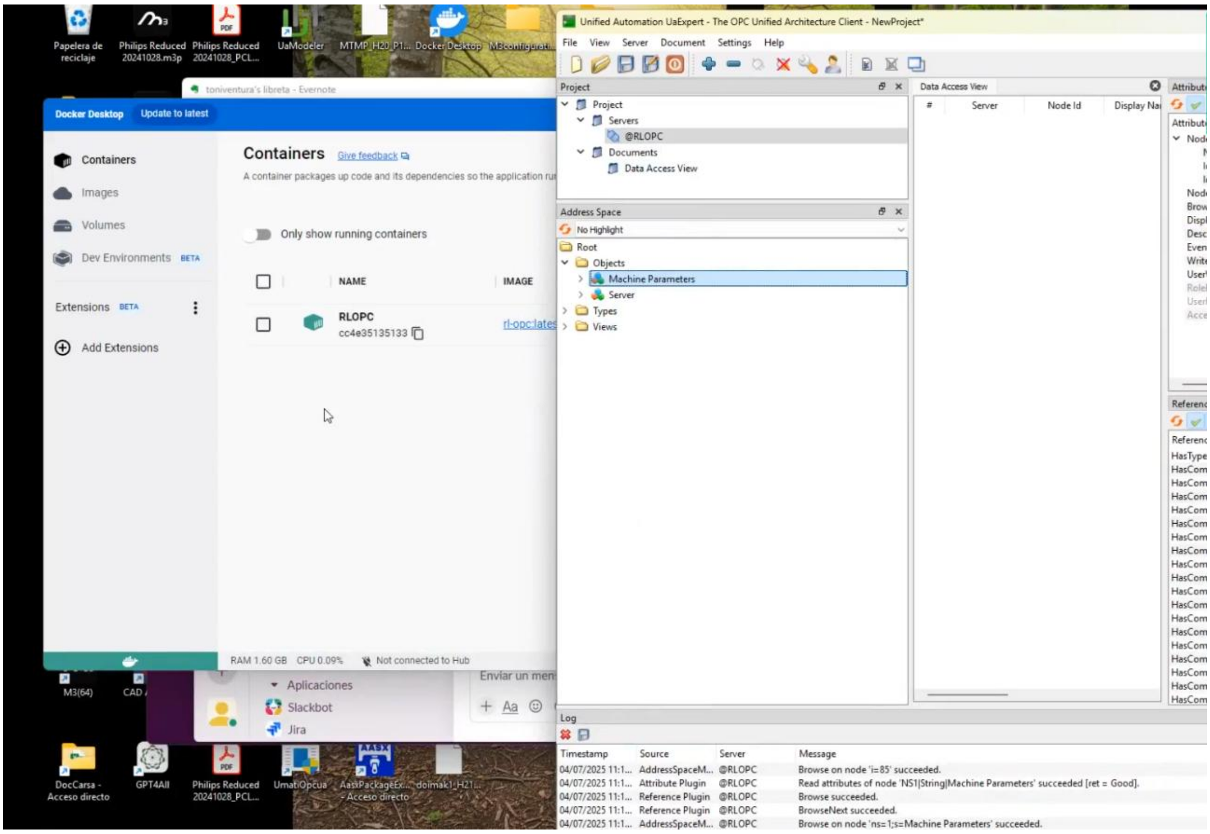


Figure 50. Pilot 1.2 RLOPC Service deployed

On the image above the connection to a OPC server can be seen (on the left background window), together with the accessible attributes, and with which the machine is working. Each of these attributes has its own characteristics and are working on real time.

aerOS can access to these machine-control attributes through (Robot Link OPC) RLOPC because a dockerized service OPCUA has been deployed, which is really what is connected to aerOS.

Similarly, from other applications such as the Human Machine Interface (HMI) or the M3 Workspace the attributes are accessible as well. These attributes can be methods (to move an axis or do a Home—go to starting position—, we can stablish the speeds of the different axis of the machine, etc). All these are running on a standardized service, which any OPCUA client may connect to access the Machine-control services remotely.

The image below showcases how we gather all information from the parameters (left side of the image), accessible via RL OPC docker service deployed. These parameters from the HMI are a combination of parameters from the hub of deployed sensors, with values from the Machine-Control system. They range from position parameters (absolute and relative to other devices), speed and acceleration on every axis, modes available for the particular machine, probe approaching angles, energy parameters (instant power, voltage) etc.

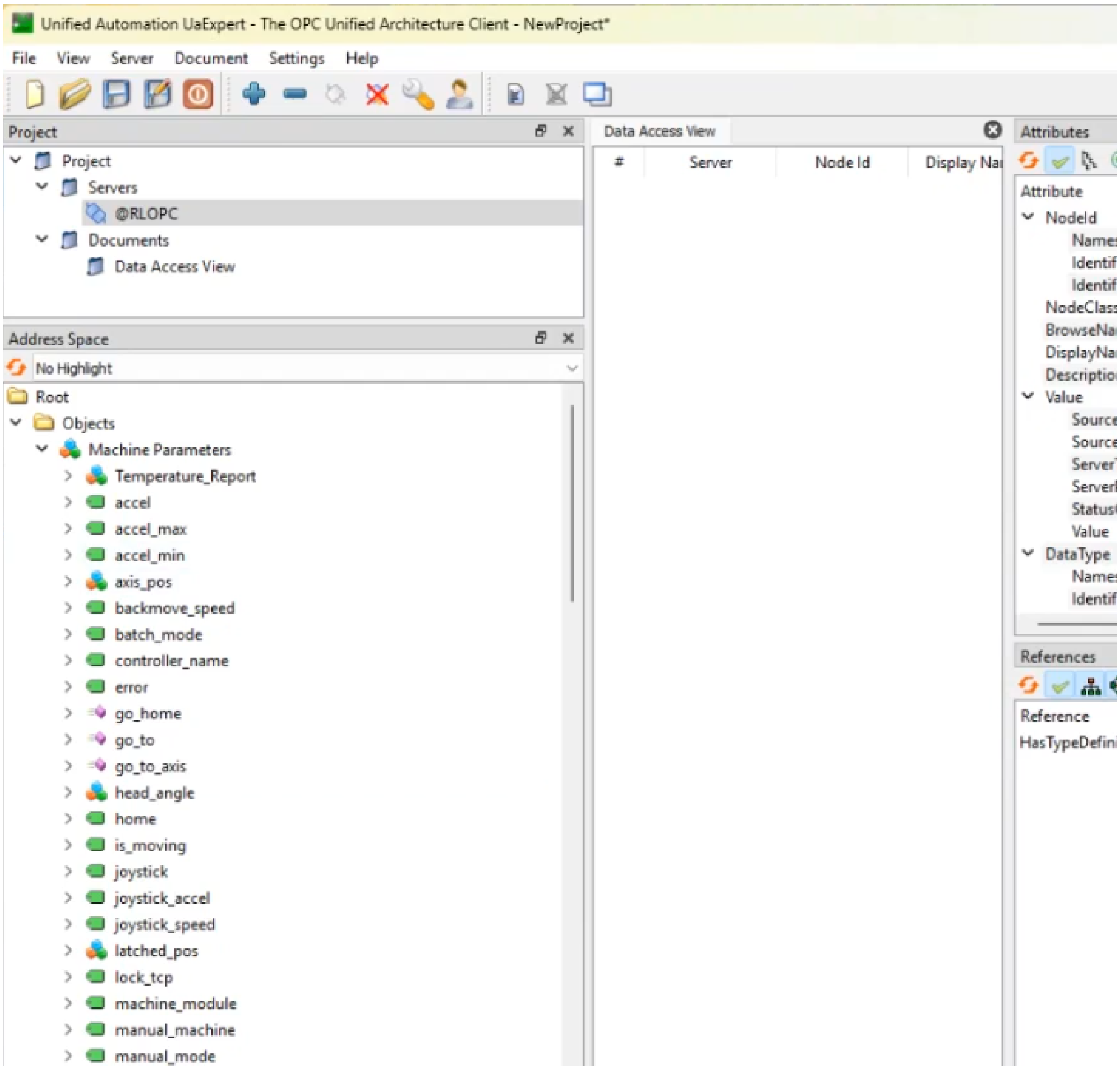


Figure 51. Pilot 1.2 List of parameters available on the OPC UA Client

2.1.2.2.3. Integration activities

Pilot 1 – Business Process 2 –Activity 6 (P1-BP2-IA6) – aerOS components:

These activities refer to the installation of the aerOS components, basic and non-basic that the pilot selected for its particular case, based on business goals and company needs.

The image below displays all the services needed on the Entry Domain (Domain 1) –please refer to the ontology diagram on section 1.1 for more information about the domains and IEs.

In descendent order, Kubernetes (K8s), Kraken-D Gateway, Federator, HLO (allocator, data-aggregator, deployment engine and frontend), Keycloak, IOTA (Api, coordinator, dashboard and hornet), management portal (Backend and Frontend), Open LDAP, Orion-LD, Self *modules (awareness, optimization, orchestrator, security, healing).

```
root@master1:/home/isantacoloma# kubectl get pods
NAME                                              READY   STATUS             RESTARTS         AGE
aeros-k8s-shim-cc767c47b-zmpc5                    1/1     Running            4 (5m32s ago)    20d
api-gateway-krakend-5c69cfd5f5-b2nr4              1/1     Running            4 (5m32s ago)    20d
federator-bc4955655-7hgxz                         1/1     Running            8 (5m11s ago)    19d
hlo-allocator-68ff66bfb8-x2vxk                    1/1     Running            4 (5m32s ago)    20d
hlo-data-aggregator-7d89c786bb-54n9q              1/1     Running            4 (5m32s ago)    20d
hlo-deployment-engine-8698885479-b9w8d            1/1     Running            4 (5m32s ago)    20d
hlo-frontend-c9bfc8fb6-vkdxq                      1/1     Running            4 (5m32s ago)    20d
idm-database-0                                    1/1     Running            7 (5m32s ago)    31d
idm-keycloak-7b675478cb-wl28j                     1/1     Running            8 (5m32s ago)    31d
iota-api-7b7647c5c9-85jqh                         1/1     Terminating        4 (5m32s ago)    20d
iota-api-7b7647c5c9-j7mcx                         1/1     Running            0                111s
iota-coordinator-bdfcdc97f-rb69l                  1/1     Running            0                111s
iota-dashboard-84f6f7dd75-flw6r                   1/1     Running            0                111s
iota-hornet-677c8d9965-lqmjf                      1/1     Running            0                112s
management-portal-backend-6867db9c4-zwsmz         1/1     Running            4 (5m32s ago)    19d
management-portal-frontend-67b6779949-nswb7       1/1     Running            6 (5m32s ago)    19d
openldap-0                                        1/1     Running            9 (5m32s ago)    31d
openldap-ltb-passwd-687ff9cd4b-lch5l              1/1     Running            32 (5m32s ago)   31d
openldap-phpldapadmin-84b7b947fd-qwhkn            1/1     Running            23 (5m32s ago)   31d
orion-ld-broker-6d5ccfdb7b-sgfr6                  1/1     Running            4 (5m32s ago)    20d
orion-ld-mongodb-0                                1/1     Running            4 (5m32s ago)    20d
self-api-self-awareness-hardwareinfo-rvzxm        0/1     Pending            0                28d
self-api-self-awareness-powerconsumptionamd64-d5x2q   0/1     Pending            0                28d
self-awareness-hardwareinfo-bllzk                 1/1     Running            12 (5m32s ago)   32d
self-awareness-powerconsumptionamd64-2tsl2        1/1     Running            11 (5m32s ago)   32d
self-optimization-api-flz2h                       1/1     Running            6 (5m32s ago)    28d
self-orchestrator-orchestrator-g4xcf              1/1     Running            7 (5m32s ago)    32d
self-security-88ld4                               3/4     CrashLoopBackOff   4286 (2m9s ago)  28d
root@master1:/home/isantacoloma# docker ps
CONTAINER ID   IMAGE                        COMMAND                  CREATED       STATUS         PORTS                                           NAMES
5a368d5ae3ff   self-healing-api             "python3 alerts_api…"    3 weeks ago   Up 6 minutes   0.0.0.0:31100->8500/tcp, [::]:31100->8500/tcp   self-healing-api
01746b03ef34   self-healing-self-healing    "python3 main.py"        3 weeks ago   Up 6 minutes                                                   self-healing-app
```

Figure 52. Pilot 1.2 aerOS components running on Domain 1

Similarly, for the second domain (Innovalia Metrology domain), which corresponds to an edge location, this is the list of installed components:

In descendent order, Kubernetes (K8s), Kraken-D Gateway, Federator, HLO (allocator, data-aggregator, deployment engine and frontend), IOTA (Api, dashboard and hornet), Orion-LD, Self *modules (awareness, orchestrator, healing). In our pilot is worth to emphasize the importance of the Self-healing module, because is the basis of the pilot ability to recover from down services.

Since the authentication is done through the Entry domain, this –Metrology domain—does not include Keycloak nor the management portal. Security and ID services connect the M3 Space with the metrology continuum, dealing with Kraken-D services managed by the aerOS MetaOS.

```
root@master2:/home/isantacoloma/self-healing/self-healing# kubectl get pods
NAME                                           READY   STATUS    RESTARTS       AGE
aeros-k8s-shim-6f59765795-vhtkr                1/1     Running   4 (10m ago)    20d
api-gateway-krakend-5c69cfd5f5-fwpk6           1/1     Running   4 (10m ago)    20d
federator-7d648d459f-n8j6p                     1/1     Running   13 (8m36s ago) 19d
hlo-allocator-68ff66bfb8-5zc7z                 1/1     Running   4 (10m ago)    20d
hlo-data-aggregator-7d89c786bb-4q45s           1/1     Running   4 (10m ago)    20d
hlo-deployment-engine-8698885479-bdrqc         1/1     Running   4 (10m ago)    20d
hlo-frontend-c9bfc8fb6-dt2gq                   1/1     Running   4 (10m ago)    20d
iota-api-7b7647c5c9-dr7rl                      1/1     Running   4 (10m ago)    20d
iota-dashboard-56684f7968-wvcsg                1/1     Running   4 (10m ago)    20d
iota-hornet-f5hsw                              2/2     Running   9 (10m ago)    20d
orion-ld-broker-854f566cb-kdmsc                1/1     Running   4 (10m ago)    20d
orion-ld-mongodb-0                             1/1     Running   4 (10m ago)    20d
self-awareness-hardwareinfo-grmnq              1/1     Running   4 (10m ago)    20d
self-awareness-powerconsumptionamd64-v69h6     1/1     Running   4 (10m ago)    20d
self-orchestrator-orchestrator-fz4tk           1/1     Running   13 (10m ago)   94d
root@master2:/home/isantacoloma/self-healing/self-healing# docker ps
CONTAINER ID   IMAGE                       COMMAND                  CREATED          STATUS          PORTS                                         NAMES
fb917e310083   self-healing-api            "python3 alerts_api…"    11 seconds ago   Up 10 seconds   0.0.0.0:8000->8500/tcp, [::]:8000->8500/tcp   self-healing-api
13956fe48d4b   self-healing-self-healing   "python3 main.py"        11 seconds ago   Up 10 seconds                                                 self-healing-app
```

Figure 53. Pilot 1.2 aerOS components running on Domain 2

Each of the ports we are using for the different services.

```
aeros-k8s-shim-service          ClusterIP   10.43.126.211   <none>        8085/TCP                                          20d
api-gateway-krakend             NodePort    10.43.181.111   <none>        8080:31449/TCP                                    20d
federator                       NodePort    10.43.86.144    <none>        8050:32228/TCP                                    19d
hlo-allocator-service           ClusterIP   10.43.234.215   <none>        8082/TCP                                          20d
hlo-data-aggregator-service     ClusterIP   10.43.20.231    <none>        8083/TCP                                          20d
hlo-fe-service                  ClusterIP   10.43.28.184    <none>        8081/TCP                                          20d
idm-database                    NodePort    10.43.99.71     <none>        5432:30790/TCP                                    31d
idm-database-headless           ClusterIP   None            <none>        5432/TCP                                          31d
idm-keycloak                    NodePort    10.43.53.91     <none>        8080:32380/TCP                                    31d
iota-api                        NodePort    10.43.148.71    <none>        6000:30634/TCP                                    5m59s
iota-dashboard                  NodePort    10.43.53.143    <none>        6060:31031/TCP,8081:31011/TCP,9311:31331/TCP      5m59s
iota-hornet                     ClusterIP   10.43.170.205   <none>        14265/TCP,9311/TCP,9029/TCP,6060/TCP              5m59s
iota-hornet-ext                 NodePort    10.43.243.10    <none>        15600:30500/TCP                                   5m59s
kubernetes                      ClusterIP   10.43.0.1       <none>        443/TCP                                           96d
management-portal-backend       NodePort    10.43.198.83    <none>        8080:31452/TCP,8082:31466/TCP                     19d
management-portal-frontend      NodePort    10.43.207.163   <none>        80:30141/TCP                                      19d
openldap                        ClusterIP   10.43.124.132   <none>        389/TCP,636/TCP                                   31d
openldap-headless               ClusterIP   None            <none>        389/TCP                                           31d
openldap-ltb-passwd             ClusterIP   10.43.229.35    <none>        80/TCP                                            31d
openldap-phpldapadmin           NodePort    10.43.179.163   <none>        80:30371/TCP                                      31d
orion-ld-broker                 NodePort    10.43.78.37     <none>        1026:31550/TCP                                    20d
orion-ld-broker-health-socket   ClusterIP   10.43.26.178    <none>        1027/TCP                                          20d
orion-ld-mongodb                ClusterIP   10.43.188.18    <none>        27017/TCP                                         20d
orion-ld-mongodb-headless       ClusterIP   None            <none>        27017/TCP                                         20d
```

Figure 54. Pilot 1.2 Active ports in the Meta OS for pilot experimentation1

Lastly, the list of users and domains accessible from the management portal, and the simplified representations of the continuum for our pilot 1.2.

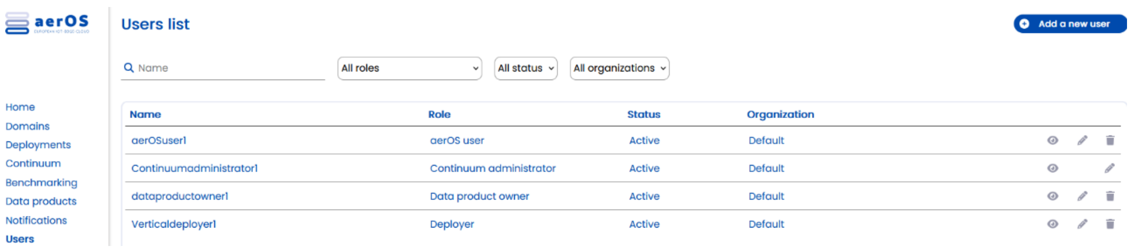


Figure 55. Pilot 1.2 Management Portal user list

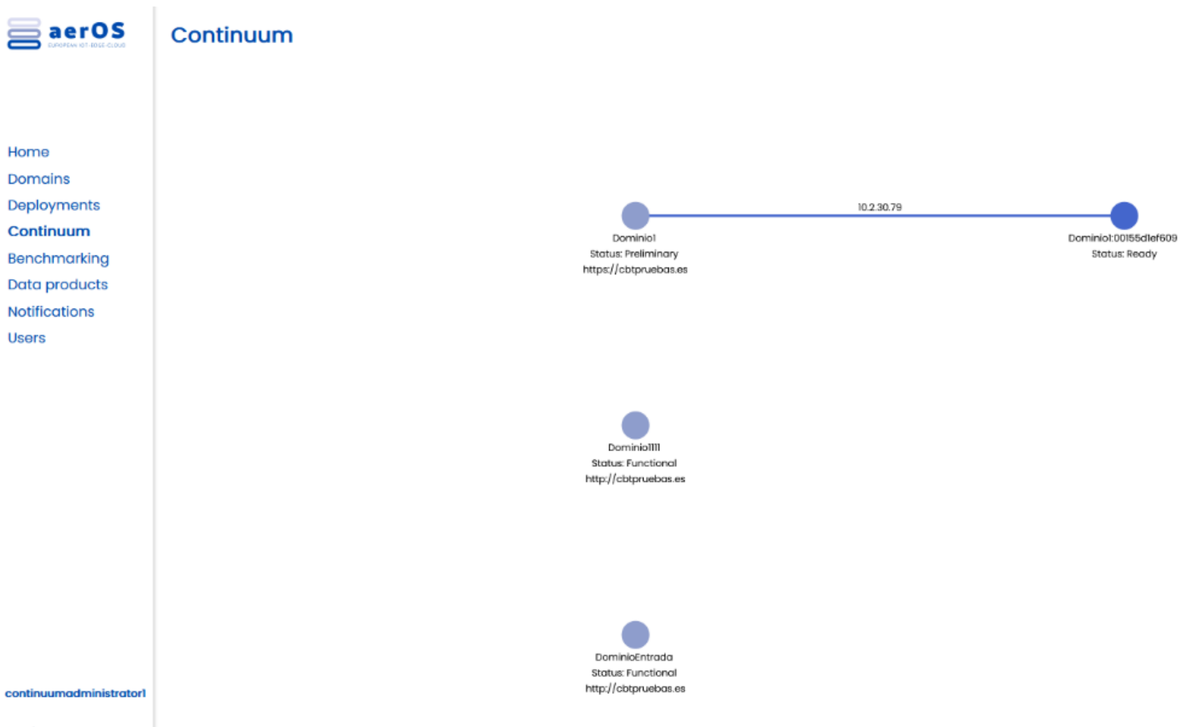


Figure 56. Pilot 1.2 Management Portal Continuum representation

By successfully integrating the aerOS components and deploying the dockerized RLOPC service (P1-BP2-DA5), we ensure MQTT and OPC-UA protocols align with the aerOS framework

2.1.3. Zero Ramp-up safe PLC reconfiguration for Lot-Size-1 Production (SIEMENS)

The scenario 1.3 of Pilot 1 demonstrates a modular, service-orchestrated production flow in which mobile assets, robot cells and edge applications are reallocated on demand. Two logical layers share the control logic:

- Behavior-tree (BT) engine – runs on the server rack in Nuremberg and determines the current step of the workflow.
- aerOS – performs every life-cycle action on the required services (deploy, start, stop, remove, redeploy) in response to BT commands.

In this scenario, frames need to be sorted by AGVs with the help of mobile robot arms. **aerOS helps orchestrating the assets and managing the life-cycle of this scenario distributed, on a software level**.

Sequence of events

1. Order initiation
    - A production request reaches the BT.
    - The BT issues an intent to aerOS; aerOS deploys and starts only the services needed for this order (navigation / ANS+, lifting, safety, ROS–TIA-Portal connector, etc.).
2. Asset relocation
    - The fleet-management component evaluates position, battery level and task queue and assigns the most suitable AGV.
    - The BT instructs that AGV to collect the required boxes from the warehouse and to move a mobile robot-arm module to the designated workstation if needed. In order to move the robot arm, a new service (skill) "lift" will need to be deployed using aerOS.
3. Sorting operation
    - Once the cell is in place, the BT tells aerOS to check which services are installed and which ones are needed to run the robot-arm application stack; the sorting routine begins.
4. Opportunistic charging
    - While the robot arm processes parts, any AGV flagged as idle is set by the BT to "charge."
    - aerOS suspends non-essential services on that vehicle—both to reduce power consumption and to avoid running minute-billed workloads; the AGV drives to its inductive charging pad, maximizing fleet availability and energy efficiency.
5. Completion and return
    - After sorting is finished, the BT signals aerOS to stop the robot-arm services—again to reduce power consumption and to avoid running minute-billed workloads.
    - Navigation and lifting services are (re)started as necessary; AGVs return all assets to their final locations.
    - The BT reaches the idle/ready state, and aerOS shuts down the remaining transient services, leaving the line prepared for the next request.

Throughout the experiment, aerOS is responsible for handling the life cycle of every containerized service, while the behavior-tree engine confines itself to deciding "what should happen next."

In D5.3, Siemens completed the groundwork for the current pilot. The team finalized the functional specification of the required aerOS components, installed four AGVs, an automatic door, a server rack and certified safety devices, and created a factory-like AGV arena in the Tech Hall equipped with wireless-charging stations. The AGVs' navigation software and their laser-scanner-based safety and collision-avoidance module were already operational, while preliminary OPC UA, MQTT and ROS 2 links interconnected PLCs, IPCs and robots, providing the fully functional baseline on which the new D5.4 developments build

During the past year, significant progress has been made in the testing production environment at the Siemens Tech Hall in Nuremberg. The infrastructure has been enhanced with one mobile robot arm module and the improvement of the AGVs, all interconnected through aerOS decentralized intelligence and SIMATIC Industrial EDGE services. A crucial advancement has been the integration of safety-certified components, including SICK NanoScan3 safety laser scanners and Siemens Safety PLCs, ensuring secure human-robot collaboration while maintaining certification standards.

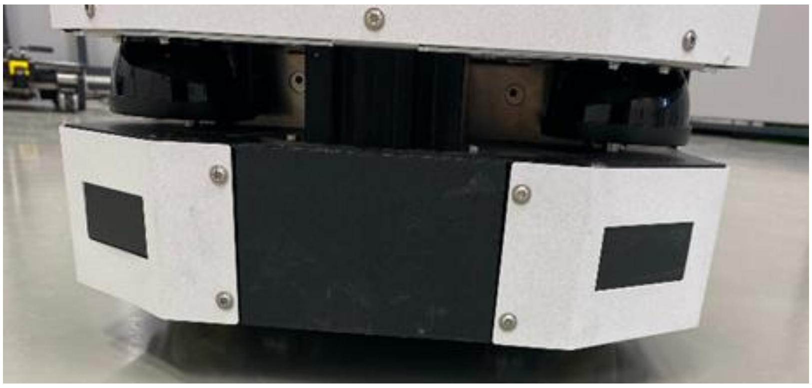


Figure 57. Photo of the laser scanners in the AGV

The system's edge processing capabilities have been substantially improved through the deployment of Industrial PCs in each AGV, each running a dedicated Software-PLC with real-time capability. Real-time in this context means that motion-control and safety functions are always completed within fixed deadlines (for example, 1–4 ms control cycles and less than 10 ms for an emergency stop). This determinism is obtained by the separation of one kernel using an hypervisor for the real-time operating system of the PLC. The integration of the aerOS core services has enhanced the ability to optimize service deployment and resource utilization on the Linux inflected part.

In a second phase, a secure VPN tunnel has been established between the Nuremberg and Munich domains, ensuring encrypted communication while maintaining strict security protocols for sensitive industrial data transmission. This secure bridge enables seamless interaction between both locations while preserving the integrity and confidentiality of operational data, while keeping consistency and integration with aerOS Meta-OS principles (and, thus components).

The Munich domain has been enhanced with the implementation of Time-Sensitive Networking (TSN) technology through a dedicated TSN switch, enabling deterministic real-time communication crucial for industrial applications.

Furthermore, a cloud domain has been integrated into the architecture, creating a comprehensive three-tier system (Nuremberg edge, Munich TSN, and cloud services). This tripartite architecture allows for extensive testing of different deployment scenarios and data flow patterns. The system can now dynamically allocate processing tasks across these three domains, optimizing resource utilization while maintaining secure and reliable communication channels.

The following table showcases how Siemens pilot has switched from processing data in the Cloud to the Edge.

Table 3. Pilot 1.3 Edge vs Cloud comparison before and after aerOS.

	Cloud	Edge
Before aerOS	No data processing in the cloud. Data requests were directly through ROS2 CLI or OPC UA.	No data processing at the edge. Data requests were directly through ROS2 CLI or OPC UA.
After aerOS	No data processing in the cloud because of SIEMENS restrictions/policies.	The Siemens Industrial Edge is running in an IPC for data vision analysis and data collec-tion.

2.1.3.1. Technical schema

The architecture of the continuum for scenario 1.3 has been strategically designed across three interconnected domains, each serving specific purposes in SIEMENS' industrial IoT ecosystem. The Nuremberg domain represents the primary edge computing environment, housing the physical automation infrastructure with four autonomous AGVs, one robotic arm and an automatic door. This domain handles real-time operations and immediate process control, demonstrating edge computing capabilities in an industrial setting.

The Munich domain functions as an experimental TSN testbed, equipped with four Raspberry Pi devices connected through a TSN-capable switch as shown in Figure 58. This setup was specifically chosen to validate deterministic communication and time synchronization capabilities in a controlled environment. The Raspberry Pi devices were selected for their flexibility and cost-effectiveness in prototyping TSN applications, while the TSN switch ensures deterministic data transmission with guaranteed latency - crucial for industrial automation scenarios.

The cloud tier remains an integral part of the three-layer architecture, but its role is carefully aligned with Siemens security policies. At this stage the cloud is used to host aerOS orchestration services and to run non sensitive components; live production data and time-critical processing continue to reside on the Nuremberg edge. **This approach lets the team exercise end-to-end deployment, monitoring, and fail-over mechanisms in a realistic multi-domain setup while keeping sensitive shop-floor data inside the Siemens network**. This three-tier architecture was deliberately designed to validate **aerOS' capability to manage distributed intelligence across different domains while maintaining secure and reliable communication through VPN tunnel**s.

This setup enables to test various scenarios of workload distribution: time-critical operations at the edge (Nuremberg), deterministic communication patterns (Munich), and non-data-processing functions in the cloud, effectively demonstrating the system's flexibility in industrial environments.

The adopted architecture for the SIEMENS Use case scenario is the schema that can be seen in the next figure.

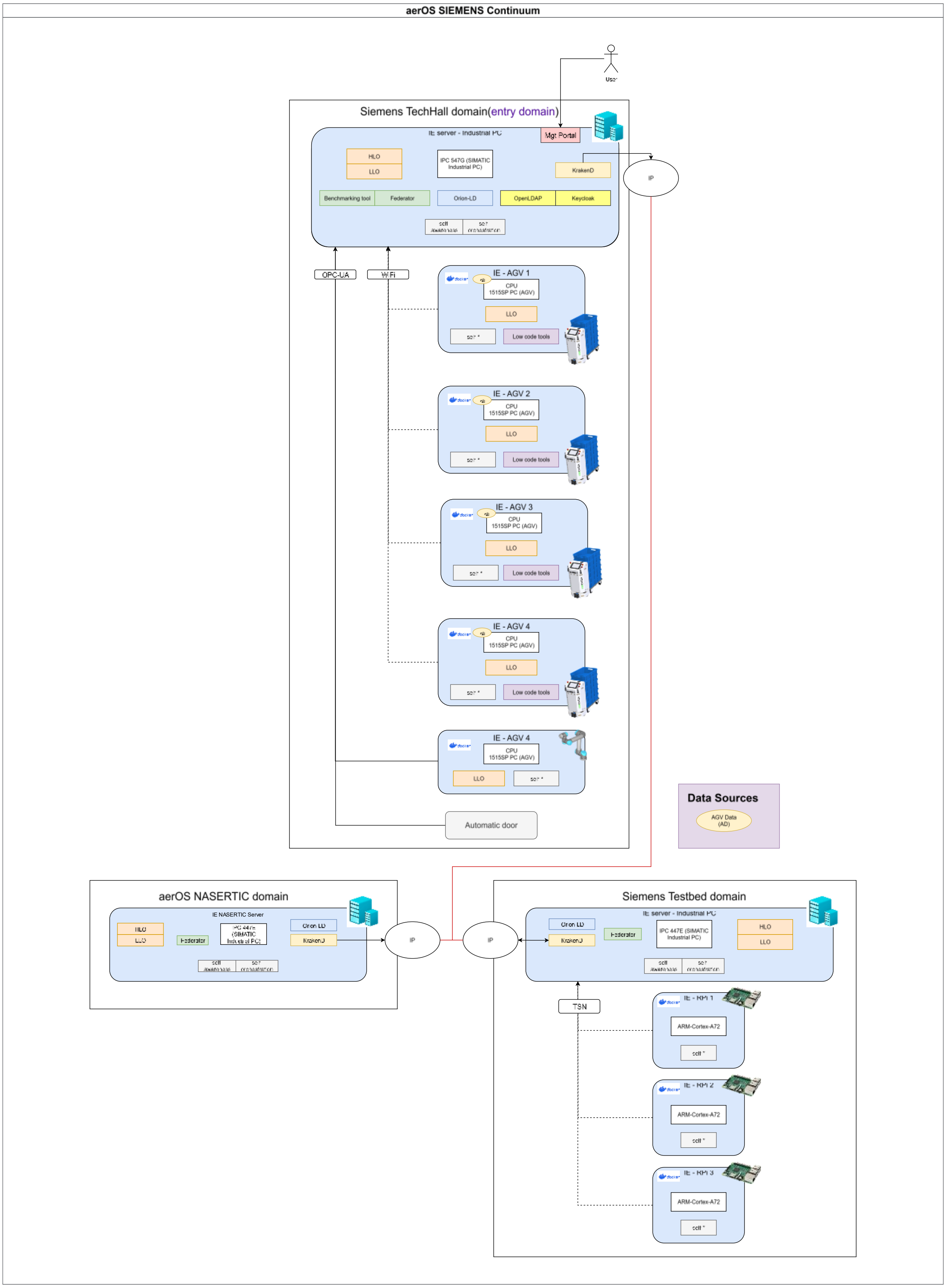


Figure 58. Pilot 1.3 Topology Diagram – Technical schema

2.1.3.2. Report of activities

2.1.3.2.1. Setup and procurement activities

To finish the setup activities, preparation of the complete set up of the robot arms and the network was required. Figure 34 proofs that the activities P1-BP3-SA3, P1-BP3-SA6, P1-BP3-SA7 and P1-BP3-SA8 have been finished.

Pilot1 – Business Process 3 – Activity - 3 (P1-BP3-SA3): Procurement and setup of 2 robot arm modules

Please refer to Figure 63. **¡Error! No se encuentra el origen de la referencia.**

Pilot1 – Business Process 3 – Activity - 5 (P1-BP3-SA5): Setup TechHall Network

A dedicated shop-floor subnet was carved out of the Siemens campus backbone and routed to the TechHall through a SCALANCE industrial firewall. Inside the hall the network is arranged in a two-level star topology.

1. **TechHall control cabinet – backbone layer**

   Acts as the main distribution frame for the TechHall. Hosts the central industrial switch, the edge server and the firewall/VPN gateway that links the hall to the wider Siemens network

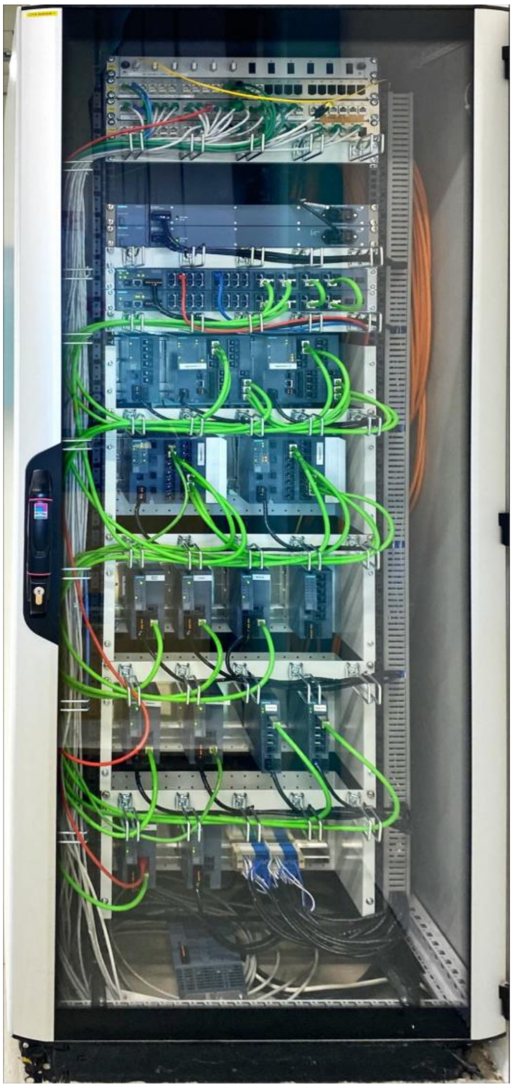


Figure 59. Pilot 1.3 Setup TechHall Network – TechHall Control Cabinet

2. AGV control cabinet – cell /aggregation layer

Sits at the edge of the arena and concentrates all connections that are specific to the mobile-robot zone.

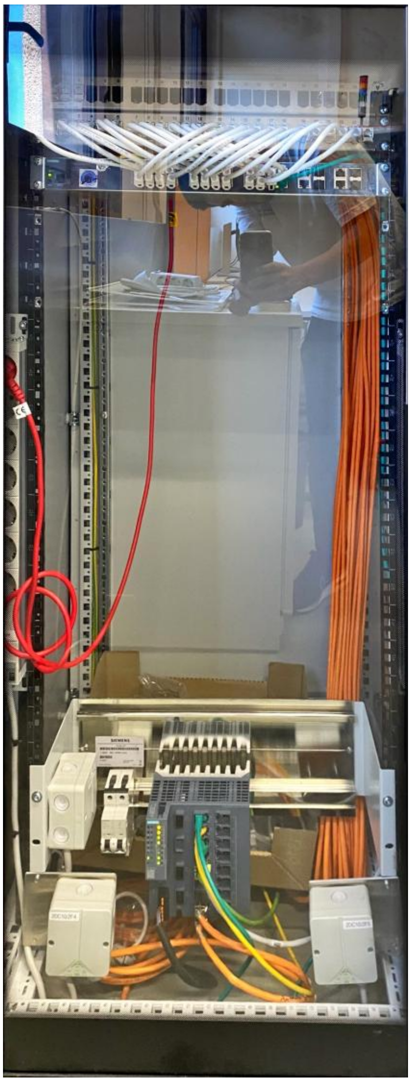


Figure 60. Pilot 1.3 Setup TechHall Network – AGV Control Cabinet

3. AGV area – field / device layer

Comprises the mobile units themselves together with the wireless access points that blanket the arena.

Figure 61. Pilot 1.3 Setup TechHall Network – Access Point for the AGVs and robot arm

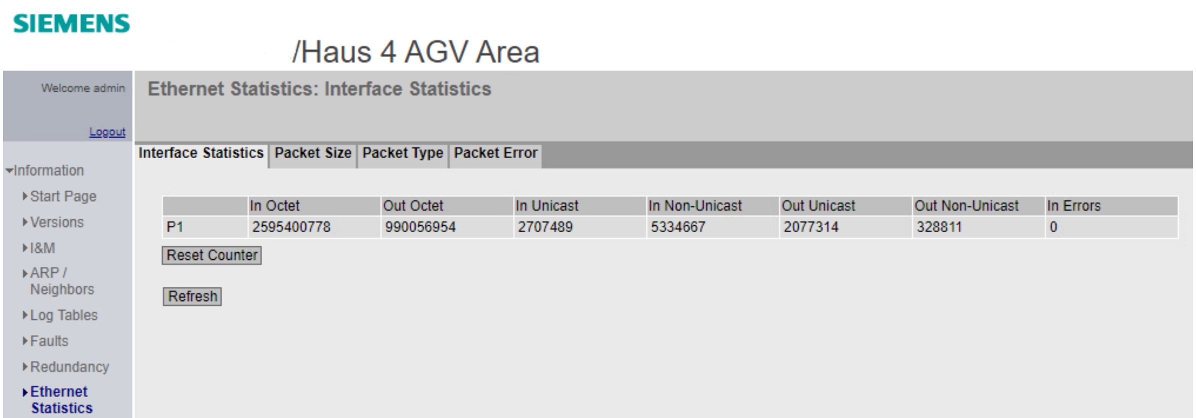


Figure 62. Pilot 1.3 Setup TechHall Network – Proof of the usage of the Access Point

Pilot1 – Business Process 3 – Activity - 6 (P1-BP3-SA6): Procurement camera for robot arm

Please refer to Figure 57.. To be able to proceed with the sorting routing, a camera is needed to recognize the products. The camera used is the RealSense D435, a stereo depth camera with up to 1280x720 pixels of resolution.

Pilot1 – Business Process 3 – Activity - 7 (P1-BP3-SA7): Setup AI camera detection

Please refer to Figure 57. The **ROS wrapper for RealSense cameras** [1]**significantly simplifies the integration of these cameras with an existing ROS system.** It acts as a crucial bridge, abstracting away the complexities of the RealSense SDK and publishing the camera's data (such as depth, RGB, and IMU streams) directly onto standard ROS topics. This allows developers to easily subscribe to and utilize the camera's output within their ROS applications. In this demonstrator, a YOLO11 model is used to detect the products that need to be sorted. This was developed under aerOS; compliant with the Meta OS and the goal of the pilot.

Pilot1 – Business Process 3 – Activity - 8 (P1-BP3-SA8): Procurement and setup gripper for robot arm

As with the camera, a gripper is required to proceed with the sorting routine. This gripper (robotiq 2F-140) is mounted on the robot arm located in Nuremberg. Crucially, the gripper's operation is also controlled using ROS 2, which allows for seamless integration and robust communication within the overall robotic system.

Figure 63. Pilot 1.3 Robot Arm with gripper and AI camera

2.1.3.2.1. Development activities

Pilot1 - Business Process 3 - Activity - 2 (P1-BP3-DA2): AGV Fleet Manager

As mentioned in the description of the demonstration, a fleet manager is used in order to manage which AGV should be in charge of the task depending on the requirements, the availability of paths and the position of each AGV. For that, the Siemens Fleet Manager has been configured and used. As can be seen in the following figures, the details of each AGV have been filled, the map of the TechHall has been created and the possible paths for the AGVs have been defined. With all this information, the software can be manually or automatically used to start a task, then the AGV that will perform the task and the best path to do it will be selected.

[1] https://github.com/IntelRealSense/realsense-ros

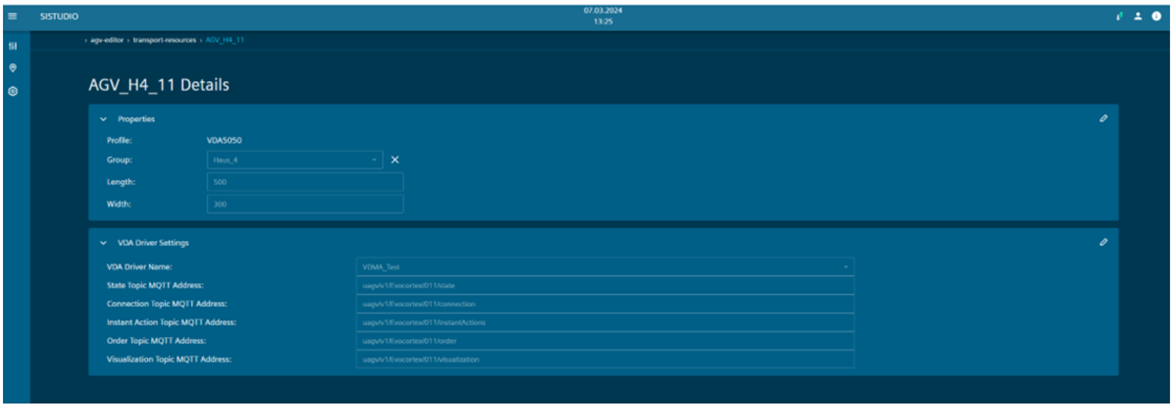


Figure 64. Pilot 1.3 AGV Fleet Managert

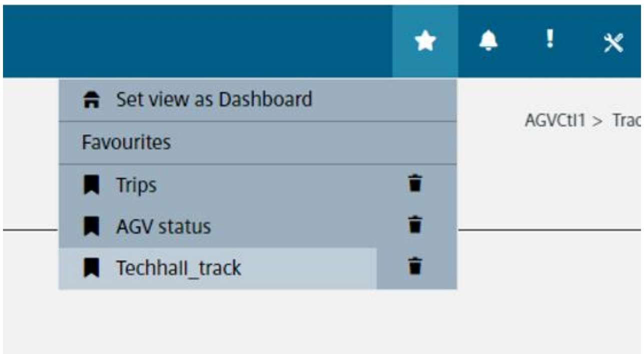


Figure 65. Pilot 1.3 AGV Fleet Manager 2

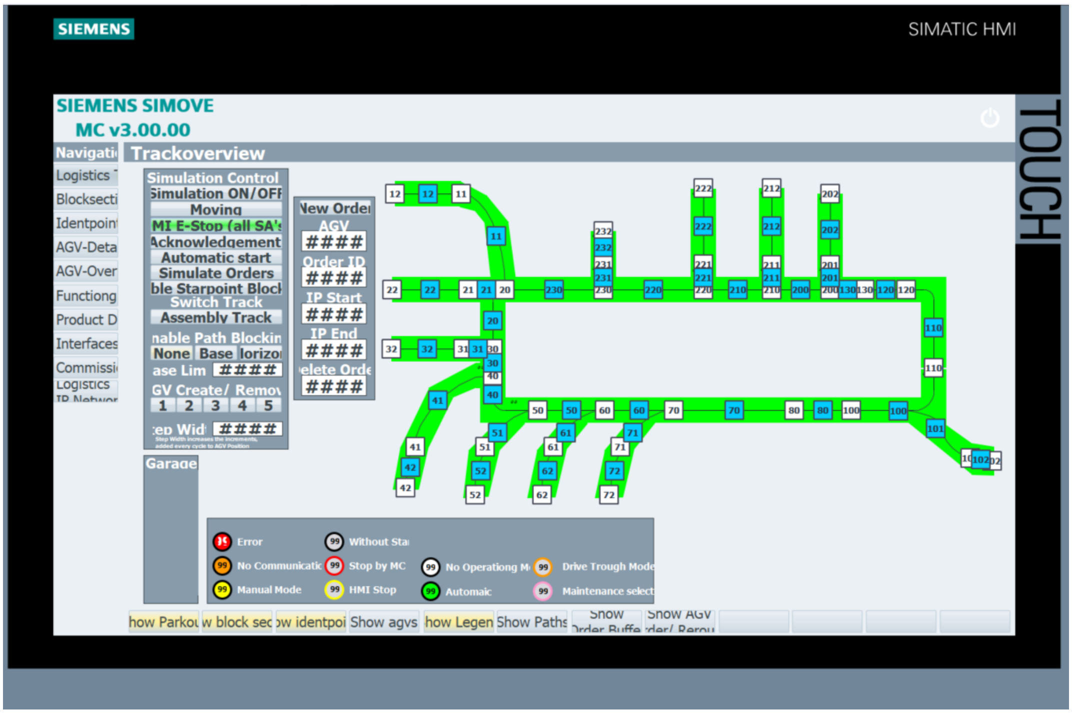


Figure 66. Pilot 1.3 AGV Fleet Manger 3

Pilot1 - Business Process 3 - Activity - 3 (P1-BP3-DA3): Device simplified programming over low code tools

To coordinate the flexible-production demonstrator, aerOS low-code tools are employed. As mentioned in the description of the experiment, in this pilot the Behavior Tree (BT) low code tool is used. The editor allows process engineers to assemble a complete shop-floor workflow by drag-and-drop instead of writing code. Each graphical node represents a predefined action—sending a command to an AGV (move. Lift, etc.), waiting for a sensor event, or branching on an execution resulting so the entire sequence from asset relocation to opportunistic charging can be expressed in a single tree.

During execution the runtime translates every node into different OPC-UA, MQTT, ROS2 requests, depending on the requirements. For example, moveAGV nodes send an MQTT message so the Siemens navigation software (ANS+) starts the movement of the AGV to the required position.

The tool is indispensable for the experiment because the scenario demands rapid reconfiguration: engineers can rearrange, insert or remove production steps in minutes and redeploy the updated logic without interrupting the underlying automation programs.

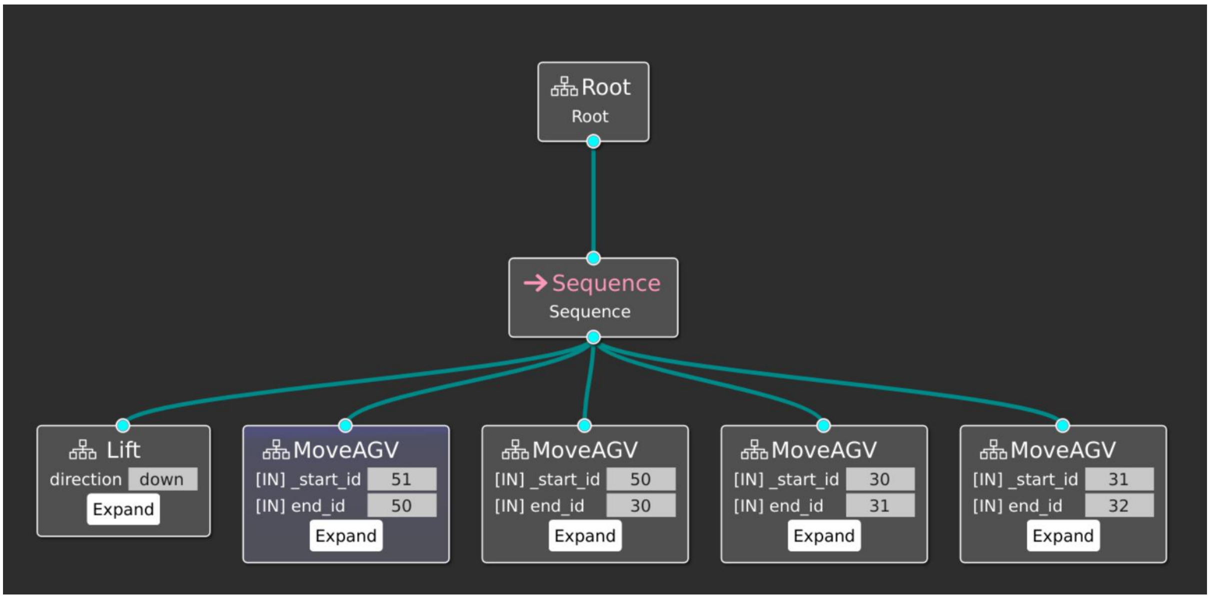


Figure 67. Pilot 1.3 Simplified programming over Low Code Tools 1

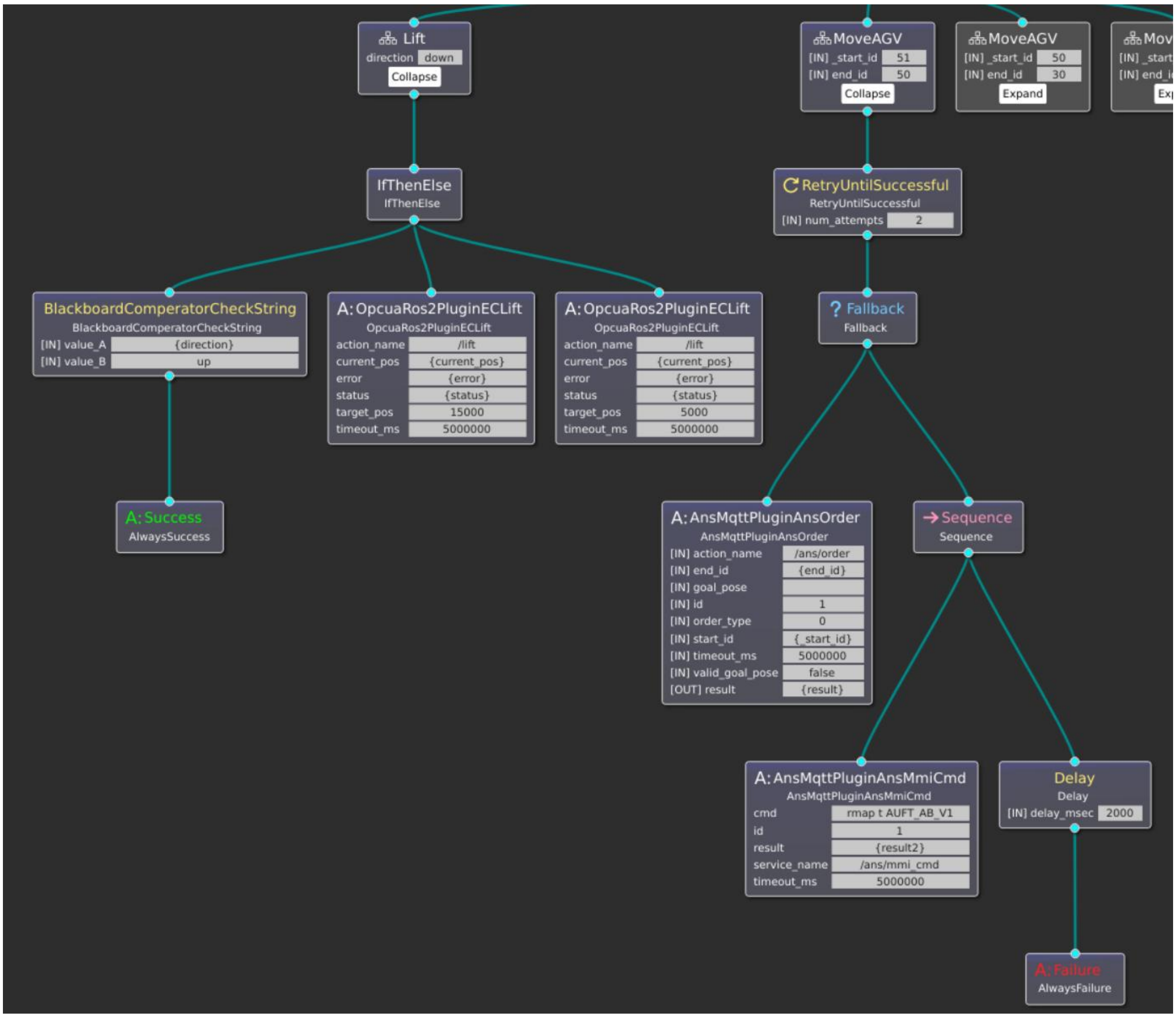


Figure 68. Pilot 1.3 Simplified programming over Low Code Tools 2

Pilot1 - Business Process 3 - Activity - 5 (P1-BP3-DA5): Siemens Industrial Edge configuration

Devices compatible with Siemens Industrial Edge already configured. The AGVs have inside the industrial PC SIMATIC IPC 127E, and the server rack is the SIMATIC IPC 547G. All of them have the Siemens Industrial OS, a hardened Linux-based operating system pre-installed on SIMATIC IPCs and Industrial Edge devices. The distribution combines a real-time–capable kernel, long-term-support packages and Siemens security extensions

Figure 69. Pilot 1.3 Device compatible with Siemens Industrial Edge – SIMATIC IPC 127E

Figure 70. Pilot 1.3 Device compatible with Siemens Industrial Edge – SIMATIC IPC 547G

Pilot1 - Business Process 3 - Activity - 6 (P1-BP3-DA6): Integration of TSN in a domain

For the use case, a Time-Sensitive Networking (TSN) switch was installed to minimise communication latency and jitter and thus improve overall system performance. The implementation uses the SoCe Multiport Time-Sensitive Networking (MTSN) Kit, a switch that supports scheduled, reserved and best-effort traffic and provides both Time-Aware Shaper and Credit-Based Shaper functions.

For the use case, a Time-Sensitive Networking (TSN) switch was installed to minimize communication latency and jitter and thus improve overall system performance. The implementation uses the SoCe Multiport Time-Sensitive Networking (MTSN) Kit, a switch that supports scheduled, reserved and best-effort traffic and provides both Time-Aware Shaper and Credit-Based Shaper functions.

The experimental arrangement (Figure 71**¡Error! No se encuentra el origen de la referencia.**) consists of a SIMATIC Microbox PC and two Raspberry Pi 4 B nodes, all linked to the MTSN Kit, which in turn is connected to a router that forwards external traffic for the scenario.

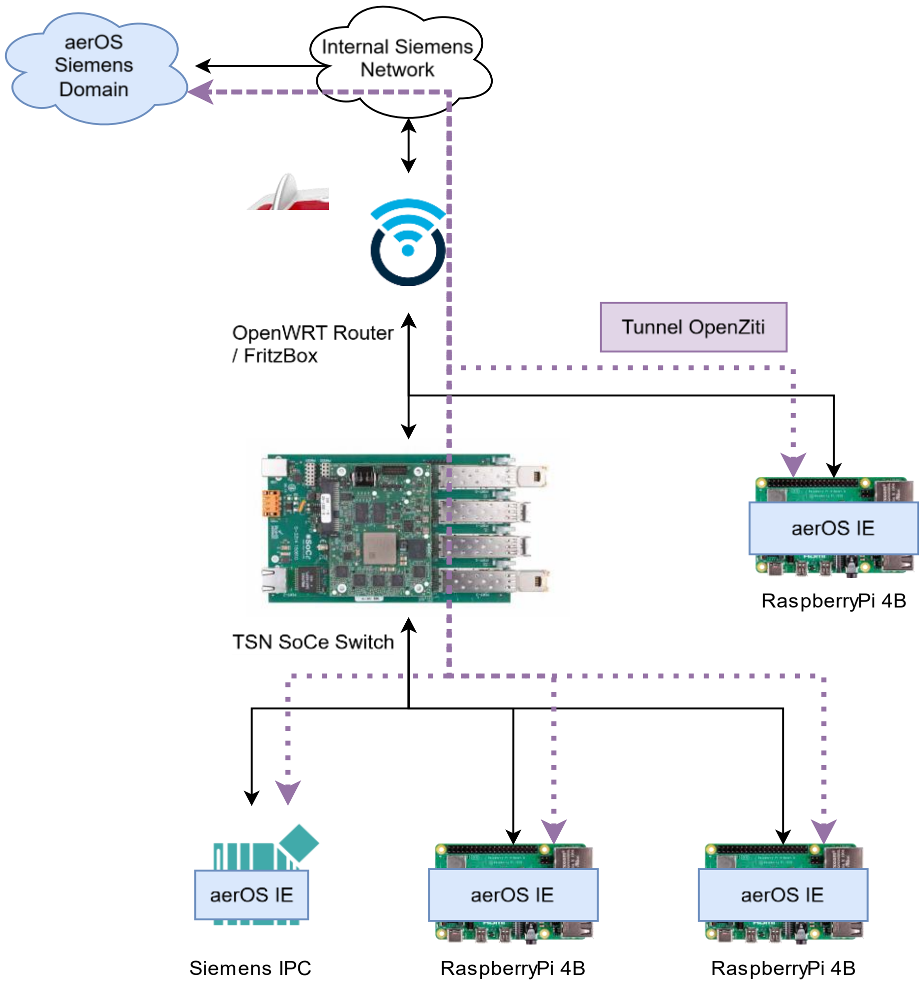


Figure 71. Pilot 1.3 Connection diagram of TSN swich integrated in aerOS

A web-based administration interface allows configuration and real-time adjustment of traffic parameters.

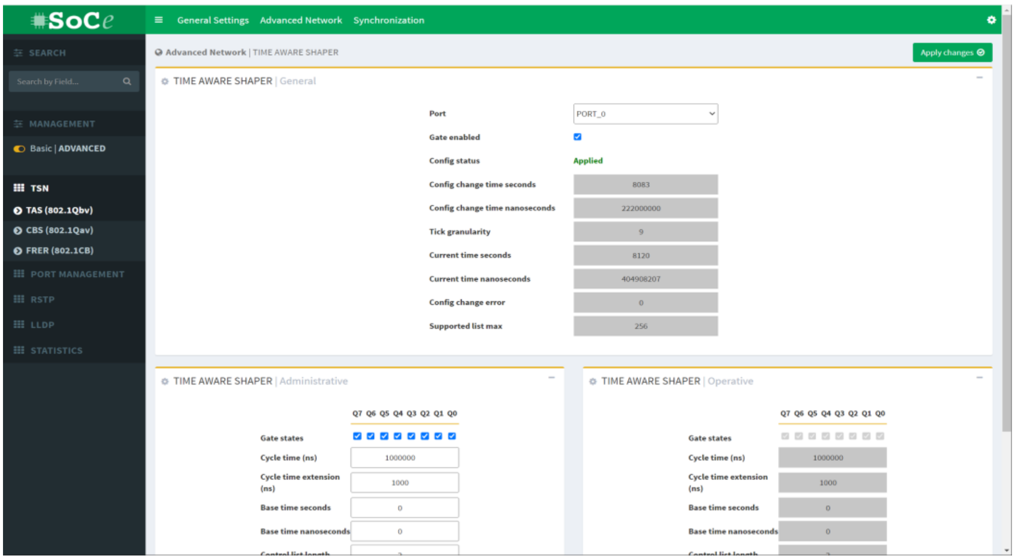


Figure 72. TSN web-based administration interface

As illustrated in Figure 73, the IPC and the two Raspberry Pis are wired directly to the TSN switch; the switch communicates with a FritzBox running OpenWRT, which supplies access to the internal Siemens network. In addition, the IPC and the Raspberry Pis maintain a secure connection to the main Siemens aerOS domain through an OpenZiti tunnel.

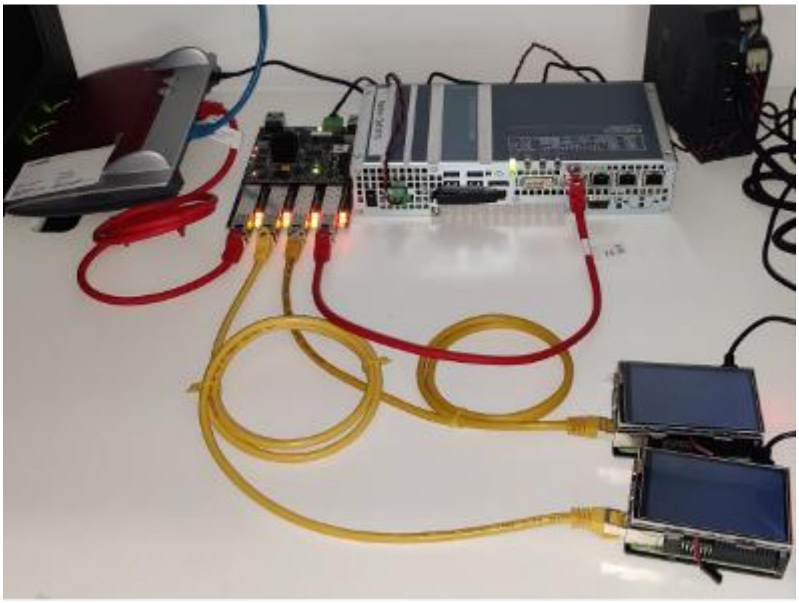


Figure 73. Integration of TSN in the pilot 1.3

2.1.3.2.2. Integration activities

Pilot1 – Business Process 3 – Activity - 2 (P1-BP1-IA2): Custom LLO for Siemens Industrial Edge device

Siemens Industrial Edge is the company's shop-floor IT/OT platform. By bringing container-based applications directly onto controllers, HMIs, IPCs and virtualised nodes, the ecosystem enables local data processing, reductions in latency-related costs, and higher overall equipment availability. Device fleets and the associated software are administered centrally, while an online Marketplace supplies ready-made or partner-developed applications.

More in depth, the Industrial Edge ecosystem consists of an Industrial Edge Hub, an Industrial Edge Manager and Industrial Edge devices. The entire ecosystem functions as a cohesive unit to streamline industrial operations:

- Application Acquisition (Hub to Management): Users or developers access the Industrial Edge Hub to find, purchase, or develop the necessary edge applications. These applications are then made available within the Industrial Edge Management system.
- Deployment and Orchestration (Management to Devices): From the central Industrial Edge Management (IEM) platform, operators can select specific applications from their library and securely deploy them to designated Industrial Edge Devices on the shop floor. IEM handles the entire deployment process, ensuring compatibility and proper configuration.
- Data Processing and Action (Devices): Once deployed, the applications run directly on the Industrial Edge Devices. These devices collect raw data from machines and sensors, process it locally according to the application's logic (e.g., performing analytics, running AI models), and then execute actions or send processed data to higher-level systems (like MES, ERP, or cloud platforms).
- Monitoring and Control (Devices to Management): The Industrial Edge Devices continuously report their status, performance metrics, and application health back to the Industrial Edge Management system. This allows administrators to have a comprehensive overview of their edge infrastructure, troubleshoot issues remotely, and ensure smooth operation.

To incorporate the Industrial Edge in aerOS, an Industrial Edge Low-Level Orchestrator (LLO) is created. This custom LLO acts as wrapper around the Industrial Edge Management API to deploy, move, or uninstall applications on the Industrial Edge.

The custom LLO, implemented in Python and containerized with Docker, exposes an API using FastAPI like the aerOS Kubernetes (k8s) LLO.

The API of Pilot 1.3's custom LLO is shown in Figure 74.

However, specific limitations with the Industrial Edge Management API need to be considered. Services that are uploaded to the Industrial Edge Management are preconfigured. On deployment, only the service and device identifier can be defined. Ports, environment variables, and command line interface arguments cannot be changed. Thus, the custom LLO processes PATCH requests by performing a full delete-and-redeploy cycle of the service. Furthermore, the service name is not unique and multiple services with the same name can be deployed on multiple devices. As a result, it is assumed that a service is installed only once, and DELETE requests are handled by the custom LLO by removing the application from all deployed devices.

The Industrial Edge instance together with the custom LLO is fully operational in stand-alone mode and has been validated against the IEM API. Integration with the wider aerOS control plane is still pending; until that is finalized, the LLO can be exercised only through direct calls rather than via the aerOS HLO.

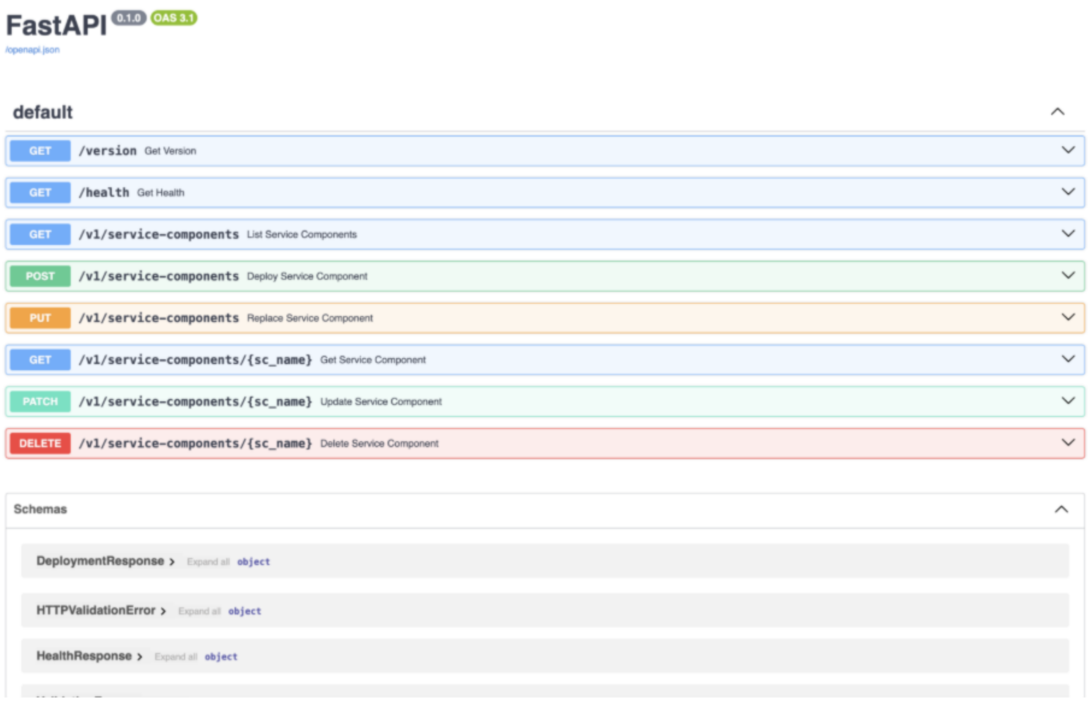


Figure 74. API of the custom LLO for Siemens Industrial Edge

Pilot1 – Business Process 3 – Activity - 3 (P1-BP1-IA3): Communicate domains using open ziti

OpenZiti[2] is an open-source, zero-trust overlay network that provides secure, identity-based access to applications and services. It makes applications invisible and inaccessible to unauthorized users, significantly reducing the attack surface by hiding services from both the public internet and internal networks.

Key paradigm shift: Traditional VPNs grant access to entire networks, exposing all connected resources once a user authenticates. OpenZiti fundamentally changes this model by providing access only to specific services. Instead of opening a broad network tunnel, it creates micro-tunnels to individual applications, ensuring users can only reach the exact resources they're authorized to use—nothing more.

Key features include:

- **Granular access control**: Define policies granting access to specific applications, services, ports, and protocols for individual users or devices
- **Secure overlay network**: Creates encrypted connections across cloud providers, on-premises data centers, and remote locations
- **Zero-trust alignment**: Enforces least-privilege access, ensuring users and devices only access resources necessary for their function

For organizations like Siemens with numerous applications, OpenZiti drastically reduces service discoverability, making them invisible to potential attackers while enabling precise access control enforcement.

[2] OpenZiti: https://openziti.io/

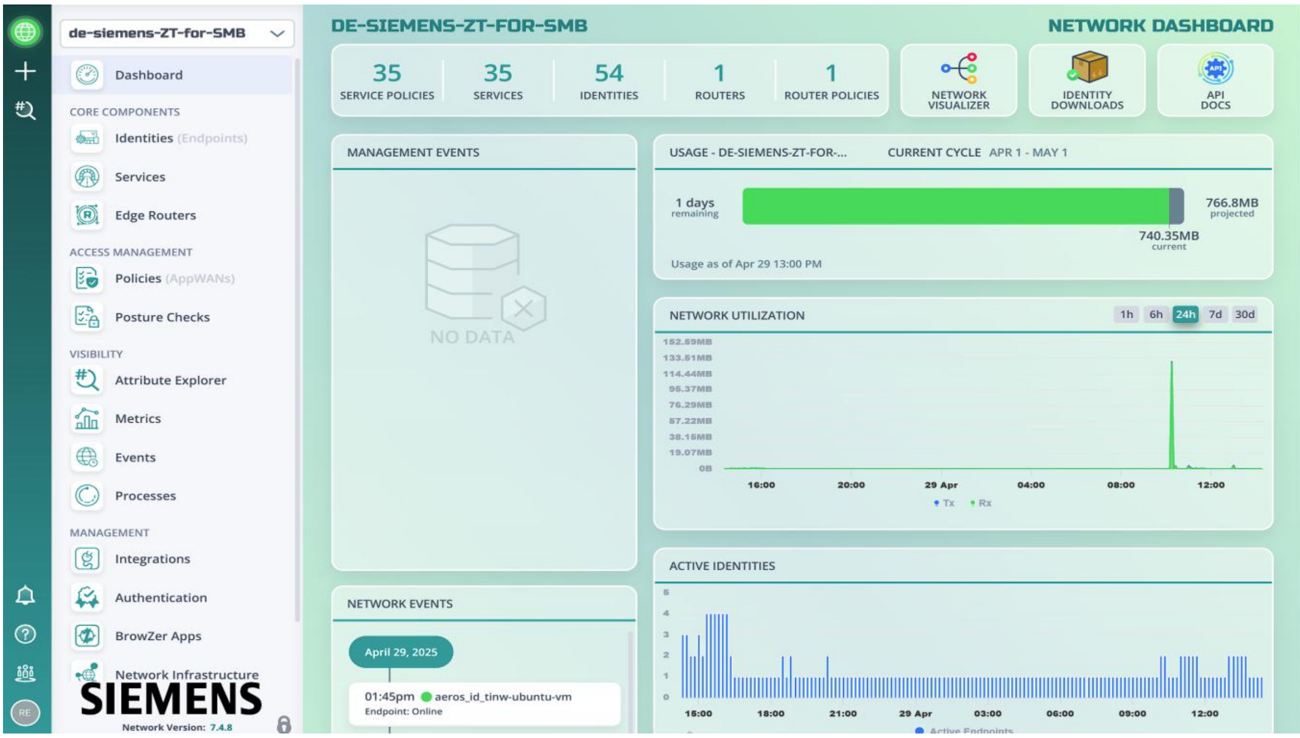


Figure 75. OpenZiti Dashboard

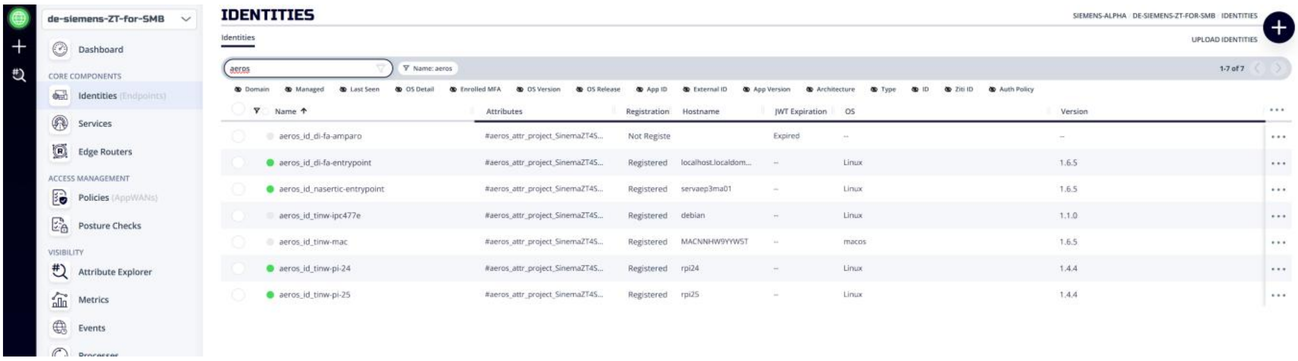


Figure 76. OpenZiti identities

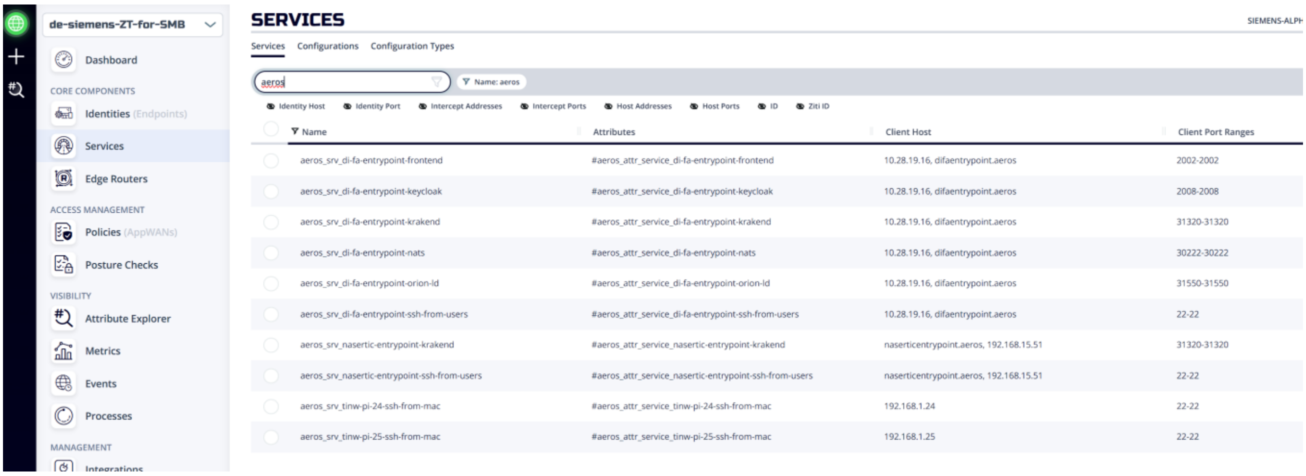


Figure 77. OpenZiti services

OpenZiti Dashboard: This screenshot displays the OpenZiti management dashboard, providing a comprehensive overview of network identities, configured services, real-time network usage metrics, and connection states. The dashboard serves as the central control point for monitoring and managing the zero-trust overlay network.

¡Error! No se encuentra el origen de la referencia.Identity Management: This view presents the identity configuration for aerOS deployments across multiple Siemens locations. The identities follow a device-centric model:

- **Domain Entrypoints**: "aeros_id_di-fa-entrypoint" (Nuremberg site) and "aeros_id_nasertic-entrypoint" (NASERTIC site) serve as primary access points for their respective Siemens domains
- **Infrastructure Elements**: Three aerOS-enabled devices ("aeros_id_tinw-ipc477e", "aeros_id_tinw-pi-24", "aeros_id_tinw-pi-25") connected to the TSN laboratory in Munich
- **Test Infrastructure**: Additional identities configured for configuration, development, and testing to aerOS domains

¡Error! No se encuentra el origen de la referencia.Service Configuration: This screen details the service exposure strategy, demonstrating OpenZiti's granular access control:

- **Application Services**: Frontend, Keycloak (authentication), Krakend (API gateway), and Orion-LD (context broker) from the primary domain, plus Krakend from the secondary domain
- **Administrative Access**: SSH services configured with strict identity-based restrictions (e.g., "aeros_srv_tinw-pi-24-ssh-from-mac" and "aeros_srv_tinw-pi-25-ssh-from-mac" accessible only from authorized notebooks)
- **Zero-Trust Implementation**: All services remain invisible to unauthorized identities, with access granted only to specific device/user combinations as defined by security policies.

Pilot1 – Business Process 3 – Activity - 4 (P1-BP3-IA4): aerOS Basic components

Installation of the aerOS basic components in Siemens domain.

```
ipc@localhost:~$ kubectl get pods
NAME                                                          READY   STATUS    RESTARTS         AGE
aeros-k8s-shim-cfbb88d78-2jpjj                                1/1     Running   5 (2d4h ago)     13d
aeros-service-5e14e86fcf09-component-nginxtest-8978f57dc-hbb9x 2/2    Running   10 (2d4h ago)    7d2h
api-gateway-krakend-7f6846997d-7mtld                          1/1     Running   5 (2d4h ago)     7d3h
benchmark-api-dfjx8                                           1/1     Running   0                7h32m
federator-7dc557d77f-xzgq9                                    1/1     Running   12 (2d4h ago)    14d
hlo-allocator-868774d7bd-qcgmc                                1/1     Running   5 (2d4h ago)     13d
hlo-data-aggregator-65c67c59c8-wjzvn                          1/1     Running   5 (2d4h ago)     13d
hlo-deployment-engine-57dd79465d-nfxf9                        1/1     Running   5 (2d4h ago)     13d
hlo-frontend-69fd64ff95-4jnm2                                 1/1     Running   5 (2d4h ago)     13d
idm-database-0                                                1/1     Running   6 (2d4h ago)     14d
idm-keycloak-778889c46d-mqw68                                 1/1     Running   6 (2d4h ago)     14d
llo-docker-operator-controllermanager-fc474d8f6-b4tbk         2/2     Running   0                5h37m
llo-k8s-controllermanager-79699c6686-45h5g                    2/2     Running   0                5h38m
llo-nats-broker-0                                             1/1     Running   0                5h50m
management-portal-backend-57cdc7dcbd-xq4pj                    1/1     Running   5 (2d4h ago)     14d
management-portal-frontend-674bbf59d5-mmf8l                   1/1     Running   8 (2d4h ago)     14d
openldap-0                                                    1/1     Running   8 (2d4h ago)     18d
openldap-ltb-passwd-d68c4cf9d-cvjd5                           1/1     Running   14 (2d4h ago)    18d
openldap-phpldapadmin-7c6b4b5666-gw5pv                        1/1     Running   21 (2d4h ago)    18d
orion-ld-broker-57dfd89d47-64xdn                              1/1     Running   17 (2d4h ago)    18d
orion-ld-mongodb-0                                            1/1     Running   12 (2d4h ago)    18d
self-awareness-hardwareinfo-wbztr                             1/1     Running   785 (2d4h ago)   18d
self-awareness-powerconsumptionamd64-wk9p6                    1/1     Running   783 (2d4h ago)   18d
self-orchestrator-orchestrator-rsr52                          1/1     Running   9                18d
wireguard-server-68c5bbd954-4jjdb                             2/2     Running   10 (2d4h ago)    11d
```

Figure 78. Pilot 1.3 aerOS basic components

Pilot1 – Business Process 3 – Activity - 5 (P1-BP1-IA5): aerOS Non-Basic components

As can be seen in the **¡Error! No se encuentra el origen de la referencia.**Figure 78, not only basic but also non-basic components have been installed for this pilot.

- Benchmarking Tool: Thanks to the installation of this tool, the network and cpu analytics can be consulted from the Management Portal like in Figure 79.

Benchmarking → Compare

Network | This is the first example domain from Siemens | Siemens01:4c5262bb3792 | 18-06-2025 15:06 - iperf 3.9 | Run a benchmark

	TCP	UDP
Media Duration	1.0000315010547638s	1.0000100016593934s
Media Bps	26714.949694240953 Mbps	1.0485655160307739 Mbps
Media RTT	60 ms	---
Media RTT var	14 ms	---
Media PMTU	65535 ms	---

Figure 79. Result after running a network benchmark

LLO Docker and LLO nats: both components were installed to be able to deploy services in the AGVs and the robot arm since they are docker based and they do not have a Kubernetes instance.

Id	Description	Public Url	Owner	Entrypoint	Status
Siemens01	This is the first example domain from Siemens	http://10.28.19.16:31320	Siemens	✓	Functional

Infrastructure elements:

Hostname	Container Technology	CPU arch	CPU cores	RAM capacity (MB)	Trust score	Status
localhost.localdomain	Kubernetes	x64	8	67213	-1	Ready
ecl4	Docker	x64	3	7065	-1	Ready

Figure 80. Example of machines that are Docker or Kubernetes-based

The EAT, Trust Score and self-security are not used in this pilot. The reason is that those components are not needed in the Siemens' use case and they are already being validated across other pilots, showcaasing their value. It was decided to focus the effort on those novel aspects of aerOS that would be required to get pilot-specific components up and running to develop the custom LLO so that Siemens Industrial Edge can be used, to further develop and use low-code technology, and to define the use case in a realistic way that ensures compliance with KPIs and the use of all the knowledge learned in this project for the future.

Pilot1 – Business Process 3 – Activity - 6 (P1-BP3-IA6): Definition of data federation – and achieving such federation

The NASERTIC domain was selected as the secondary domain to the primary site in Nuremberg for two main reasons.

1. Resilience and local access: continuous availability of the management portal must be maintained even if external Internet links to cloud services are disrupted.
2. Data-residency and security: confidential manufacturing data must remain on the shop floor in Nuremberg in compliance with internal policies.

Secure, direct communication between the two domains is provided by OpenZiti, which is installed on designated entry-point devices in both locations. The resulting zero-trust overlay supplies encrypted, bidirectional traffic without the need for extensive firewall reconfiguration.

Within the NASERTIC domain the following aerOS components have been deployed: Orion-LD Context Broker, aeros-k8s-shim, KrakenD, HLO, LLO, the self-* modules and the aerOS Federator. The Federator synchronises deployments between the primary and secondary domains; in NASERTIC the parameter isEntrypoint is set to false and peerFederatorUrl points to the OpenZiti address of the Federator entry-point in Nuremberg.

2.1.4. AGV Swarm Zero break-down logistics for Lot-Size-1 Production

Regarding scenario 1.4 (deployed by MADE and POLIMI); the previous phase (till M18), efforts were dedicated to defining the aerOS components and use case requirements, producing comprehensive technical documentation, and validating integration plans. Activities included procuring and preparing necessary equipment, initiating deployment within the POLIMI facilities, and setting up communication systems to support AGV and production line coordination. Additionally, key performance indicators (KPIs) and validation frameworks were established to support continuous analysis and evaluation of the use cases. During the previous phase, implementation activities focused on the installation and integration of the basic components of the aerOS system, along with the development and deployment of the main pilot software modules, including the Order Manager, Outsourcing module, AI/ML module, and AGV path planner.

During the period M18-M35, implementation activities have instead focused on consolidating the pilot system integration within the aerOS architecture. Key milestones achieved during this phase include:

- Integration of all domains: All pilot domains (MADE, POLIMI, NASERTIC) were successfully integrated, ensuring interoperability across the system.
- System architecture consolidation: All basics components required for the pilot architecture to function cohesively were integrated and validated.
- Application-level developments:
    - These services are primarily used for achieving The Autonomous Guided Vehicle (AGV) path planning and navigation andnd Order management.
    - Integration of Orion-LD was completed, enabling semantic data exchange within the new order management system.
- Automation and deployment: All system components were containerized and packaged, supporting automated deployment pipelines. Since the Continuum consists of a mix of ARM64 and AMD64 compute nodes all the images are dual architecture.
- Experiment conduction: The Pilot test and validation activities were carried out during this period. The activities were conducted through different steps: in the first, individual modules were tested in isolation; then, integration tests validated interoperability across all domains; finally, scenario-based pilot tests were executed to evaluate system performance under realistic operational conditions

Initial system architecture heavily relied on cloud-centric data processing, with limited/zero use of edge capabilities. Thanks to aerOS, the distribution of applications toward the edge rebalanced this distribution, aiming for a more edge-oriented processing architecture to meet latency, resilience, privacy and general efficiency requirements.

Enabled by the intelligent orchestration in aerOS Meta-OS, 12 out of 13 applications (92%) are now deployed at the edge, with only 8% relying on cloud infrastructure. The orchestration capabilities of aerOS further support the coordination of additional containerized workloads across the established continuum.

The following table showcases how our pilot has switched from processing data in the Cloud to the Edge.

Table 4. Pilot 1.4 Edge vs Cloud comparison before and after aerOS.

	Cloud	Edge
Before aerOS	All Cloud based	
After aerOS	Order Manager	@Context (MADE), Order Generator, Orion-LD (MADE), MongoDB (MADE), OM-MADE, ML Module, @Context (POLIMI), Orion-LD (POLIMI), MongoDB (POLIMI), OM-POLIMI, Button, AGV Path planner

2.1.4.1. Technical schema

The adopted architecture for the Pilot 1.4 is the schema that can be seen in the next figure.

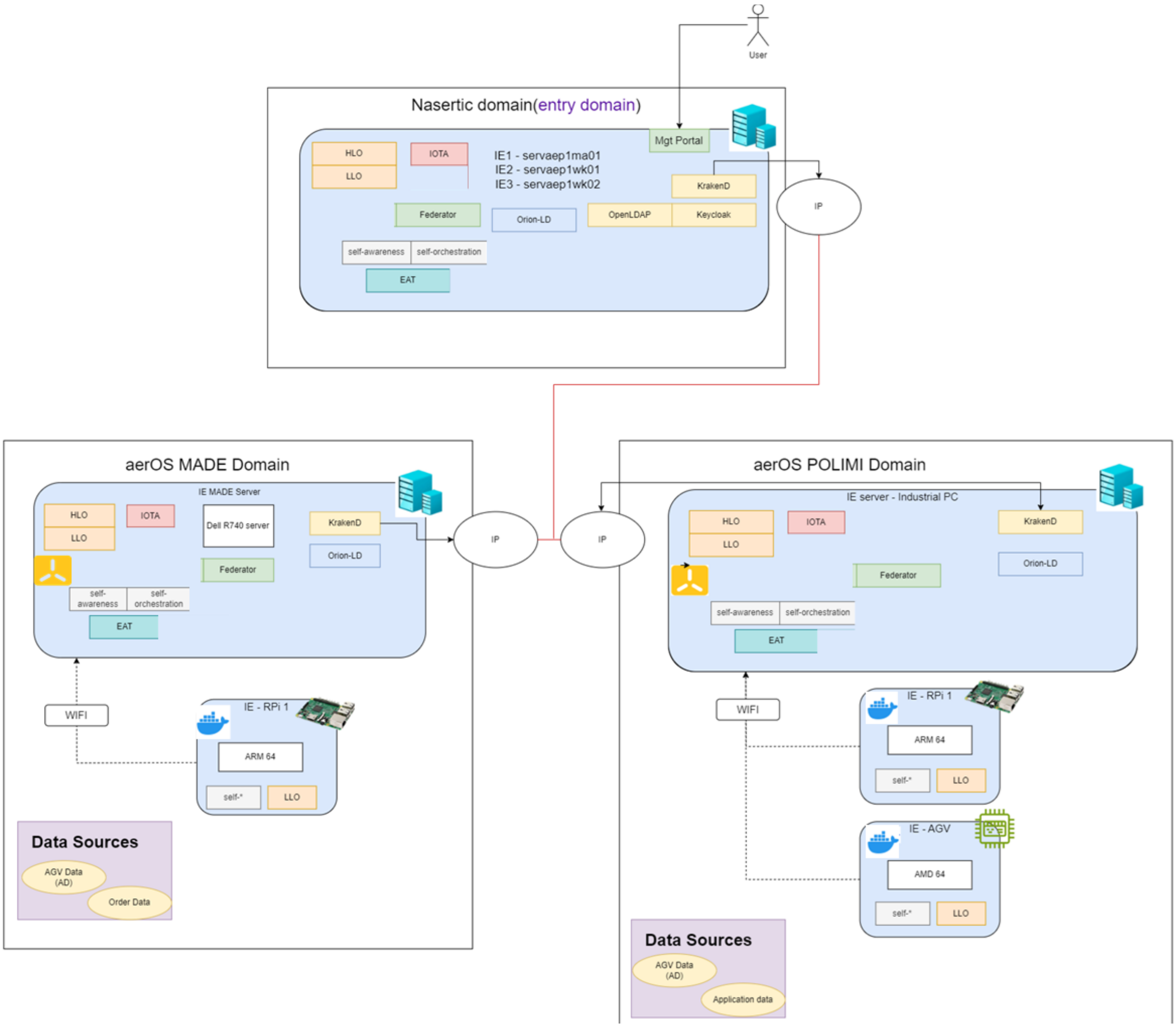


Figure 81. Pilot 1.4 Topology diagram – Technical schema

The diagram illustrates the technical setup for Pilot 1.4, differentiated into three distinct aerOS domains that support various operational tasks across different system components. Below is a breakdown of how the system is structured:

- NASERTIC Domain (Entry Domain): The NASERTIC Domain acts as the entry point into the system, providing the user management portal. This domain consists of k8s cluster with 1 master node and 2 worker nodes.
  - The management portal serves as an entry point for end users to monitor status of the different domains and eventually deploy services to different infrastructure elements. Role based access to the Management portal ensures proper accesses are provided to different user levels and for this authentication is managed by Keycloak service.
- aerOS MADE Domain:
  - The MADE Domain serves as the first manufacturing factory. This domain features 1 IE Edge Server and one IE RPI (ARM 64 architecture).
  - The IE server is a single node k3s cluster and the raspberry pi uses docker.
- aerOS POLIMI Domain:

- The POLIMI Domain serves as second Manufacturing factory. It consists of Industrial PCs, Raspberry Pi device, and another edge devices from the AGV.
- On similar line with the MADE domain, the Industrial PC hosts a single node k3s cluster, and the AGV and raspberry pi runs docker.

This architecture relies on a mix of infrastructure elements with k8s, k3s and Docker which is in line to the vision of aerOS as being a platform agnostic meta operating system. The two manufacturers (MADE and POLIMI Domains) simulate an external production strategy in the case aerOS predicts that the production line will be saturated. The simulation is performed over real AGVs, deriving results that will be directly applicable over production lines similar to the ones described in the pilot.

2.1.4.2. Report of activities

2.1.4.2.1. Setup and procurement activities

The activities from P1-BP4.SA1 to P1-BP4-SA4 has been successfully executed and a more detailed description can be found of corresponding section of Setup & Procurement activities for pilot 1.4 within D5.3.

Pilot1 – Business Process 4 – Activity - 5 (P1-BP4-SA5): Setup of Domain Infrastructure

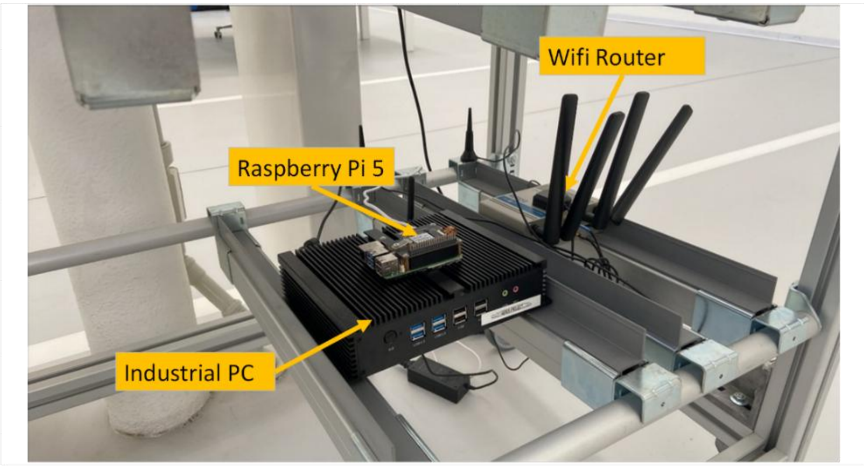


Figure 82. Pilot 1.4 Setup of Domain Infrastructure 1

The Figure above shows 2 out of the 3 infrastructure elements (i.e. Raspberry pi 5) and The industrial PC and an industrial grade WIFI-router of POLIMI Domain. In context of the aerOS architecture of Pilot 1.4, the Industrial PC acts as the IE Server and it hosts majority of the aerOS services like Orion-ld etc. in addition to the aerOS services it also hosts the Order manager of the POLIMI Domain

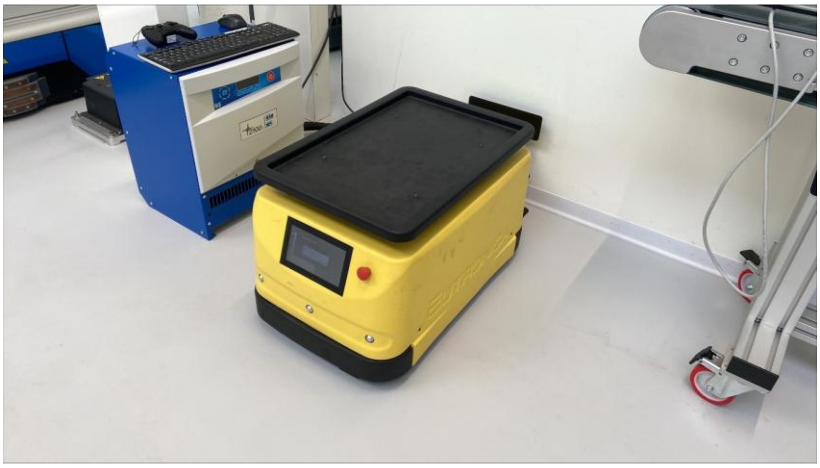


Figure 83. Pilot 1.4 Setup of Domain Infrastructure (POLIMI)

The Figure above illustrates the AGV component used in the POLIMI domain. This is an industrial AGV called Jobot, designed by Eutronica, with a maximum payload of 50 kg. The AGV runs on Ubuntu 18.04 and is developed using the ROS Noetic framework. It is equipped with an RPLIDAR 360 for mapping the surroundings and also features electromechanical microswitch bumpers, which work in conjunction with the LiDAR to avoid collisions in dynamic environments.

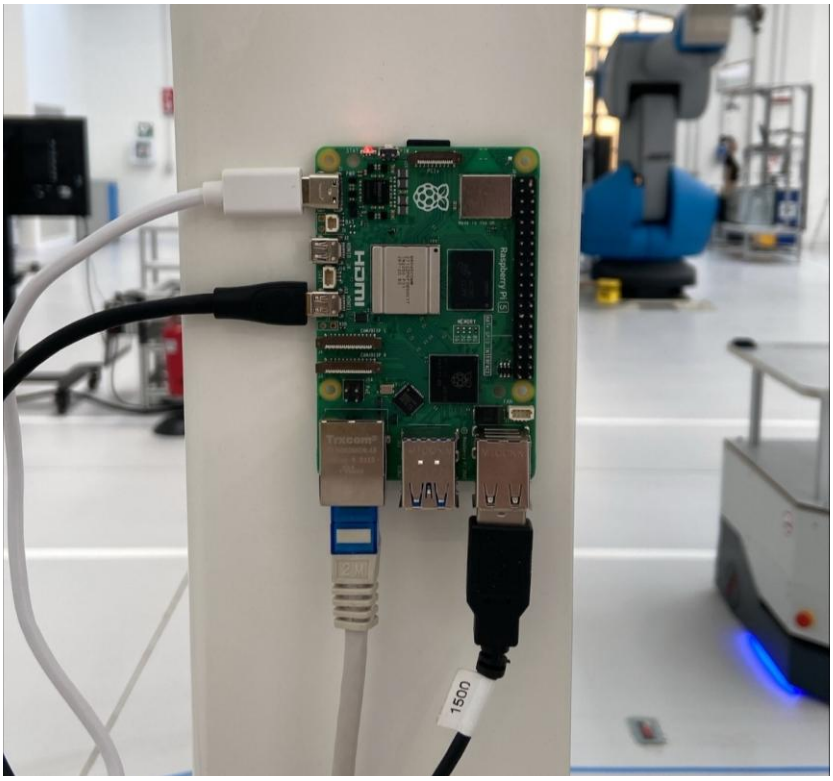


Figure 84. Pilot 1.4 ***Setup of infrastructure (Raspberry Pi 5 on MADE premises)***

Pilot1 – Business Process 4 – Activity - 6 (P1-BP4-SA6): Setup of Dedicated network for MADE Domain

This Activity has been completed successfully, and its validation can be taken from the other activities.

2.1.4.2.2. Development activities

In the POLIMI domain shown in Figure 5, the AGV Path Planner is indicated.

It receives incoming orders and, based on each order, provides the appropriate path from a set of possible paths for the AGV to follow. Furthermore, Figure 85 illustrates that the AGV server is used to receive incoming orders via a REST API and forward them to the ROS-based navigation system.

This system, taking into account the AGV's current location, available paths, and the goal destination, communicates with the AGV drivers to guide the vehicle to its target.

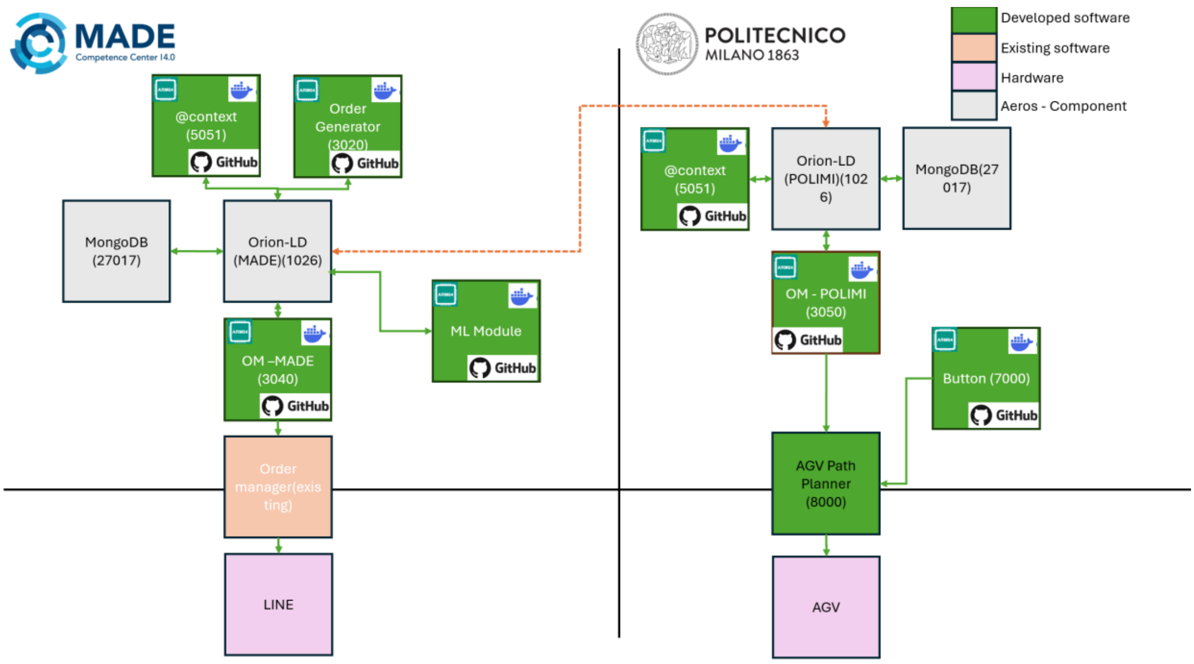


Figure 85. Pilot 1.4 Overall Use Case Application Architecture

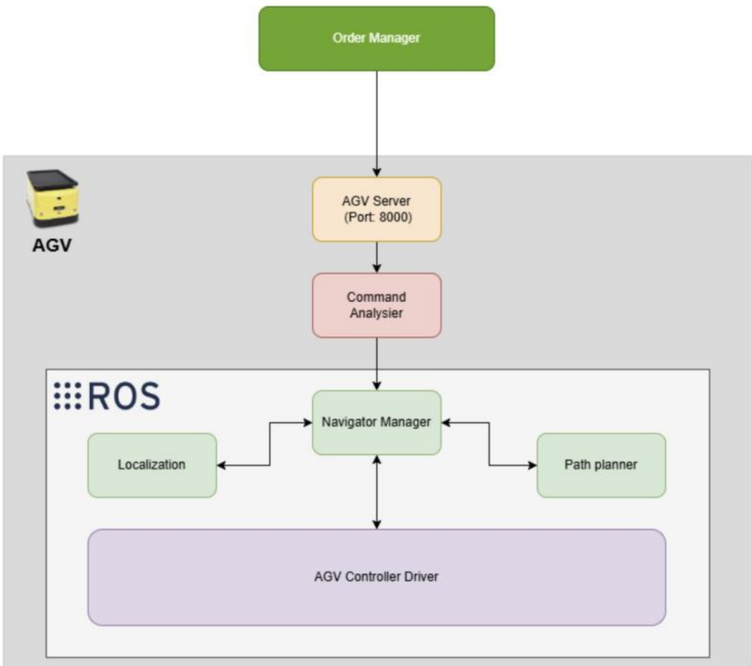


Figure 86. Pilot 1.4 AGV detailed architecture

Pilot1 - Business Process 4 - Activity - 1 (P1-BP4-DA1): Development/containerization of overlay Order management application at MADE

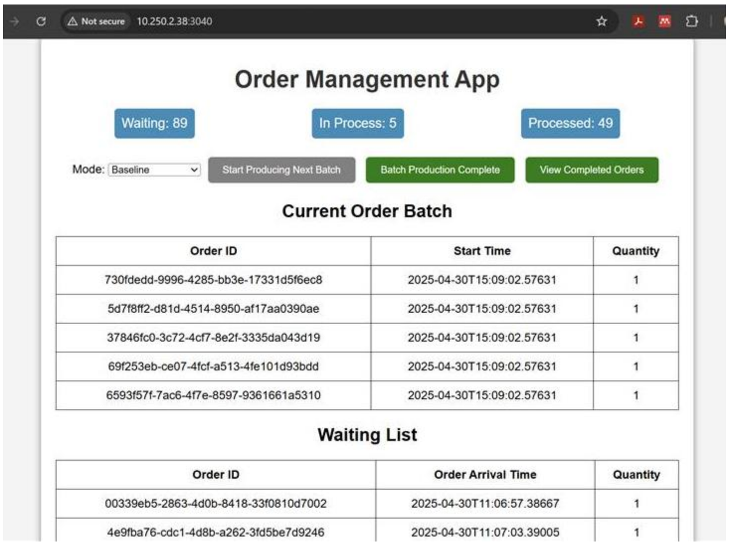


Figure 87. Order management application at MADE custom development

The order manager App is the primary front end accessible to the operator on the Production line the app allows the operator to start production cycles, view the information about orders being processed as well as orders in the queue.

Additionally, a summary of number of valves in processing, Waiting and Processed number of valves is summarized at the top. Additionally, it allows the user to view some information related to historical data of the processed orders. The Application was developed specifically in context of aerOS project and relies on Orion-LD entities for storing the information about the orders being processed.

Prior to aerOS, functionality was being accomplished by a cloud-based platform called LEA, the system was quite rigid and would start the line as soon as a new order entered the system, this was quite inefficient in terms of both energy and production efficiency particularly the number of AGV travels. This was because the system does not allow the combination of multiple orders. Therefore, even though the line is capable of handling a batch of 5 valves, such combination was not feasible with the existing system.

Duning the course of aerOS project, the order manager was developed which works like a middle agent allowing for the aggregation of the multiple orders to optimize AGV travels.

The Order Manager application was developed in Python. Since the Pilot uses both ARM 64 and AMD64 IE elements, A dual Architecture (ARM64 and AMD64) image of the app was developed and is openly available on Docker Hub3. The app was developed specifically in context of aerOS Project.

[3] https://hub.docker.com/repository/docker/danny0117/aeros-madman2/general

Pilot1 - Business Process 4 - Activity - 1 (P1-BP4-DA2): Development/containerization of overlay order management at POLIMI

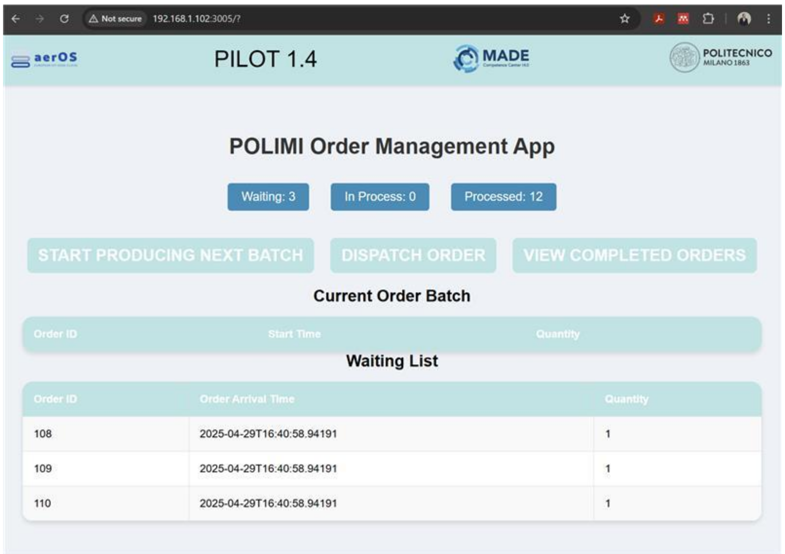


Figure 88. Order management at POLIMI

The POLIMI Side of the Order Management App is in principle similar to the MADE application when it comes to the front-end functionality, however the backend is significantly different. Whereas MADE application uses the LEA system APIs to indirectly control the production line, The POLIMI app directly interacts with the underlying AGV trough the APIs exposed through the AGV Path Planner

In line to the multi architecture compute nodes, a dual architecture (ARM64 and AMD64)[4] image was created. This app was developed specifically for aerOS Project.

Pilot1 - Business Process 4 - Activity - 1 (P1-BP4-DA3): Development of Synthetic order generator application

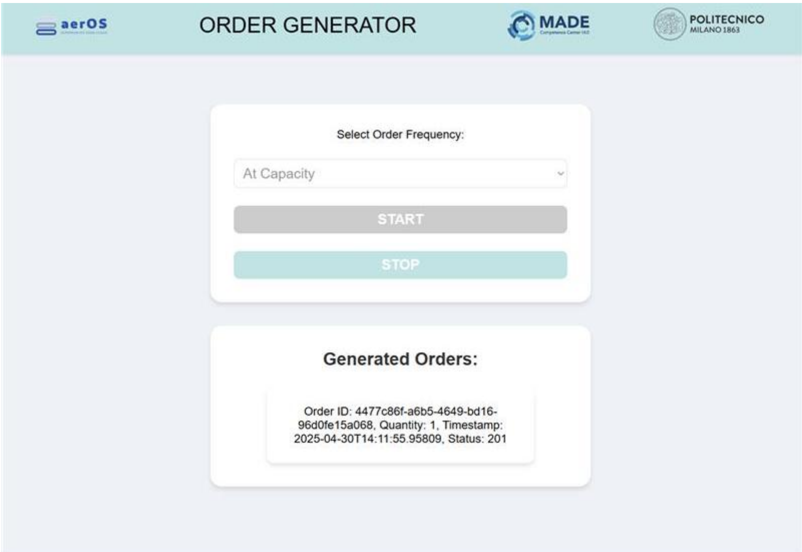


Figure 89. Pilot 1.4 Synthetic order generator application

Although the Valve manufacturing line is a fully functional working production line inside MADE cc, MADE is not a real manufacturing plant, therefore does not receive actual orders from clients, therefore the order generator app was developed to mimic the inflow of actual orders. The order generator generates orders with a

[4] https://hub.docker.com/repository/docker/danny0117/aeros-poland/general

specified mean interarrival with exponential distribution. The mean value can be changed with by selecting the Mode from the dropdown.

The Order generator was developed in Python and the orders so generated are stored in orion-ld entities of type "Order". Considering the fact that the Pilot uses both ARM 64 and AMD64 IE elements, A dual Architecture (ARM64 and AMD64) image of the app was developed and is openly available on Docker Hub[5] . The app was developed specifically in context of aerOS Project.

Pilot1 - Business Process 4 - Activity - 1 (P1-BP4-DA4): Development of order backend persistor

This activity was no longer a strict need in the pilot since the application architecture was updated and this component was replaced towards a more straightforward implementation of the use case. However, effort was put to achieve the requirements expected, that were satisfactorily met.

The initial version of the application used a single Orion-LD entity with id urn:ngsi-ld:order:order001 to store all orders in the form of a list. In this version, the order generator, instead of creating an new order entity for each new order, made a patch request to a single entity with id urn:ngsi-ld:lastOrder:lastOrder001. At setup time a subscription in the Orion-ld was created to this lastOrder entity and when a new order would be generated, a /update_order_list endpoint in the backend persistor app would append the new order to the order001 entity's orderList attribute.. In the updated system, each order is now stored as a separate entity and hence there is no need of the persistor app.

The backend persistor app was developed in Python. Considering the fact that the Pilot uses both ARM 64 and AMD64 IE elements, A dual Architecture (ARM64 and AMD64) image of the app was developed and is openly available on Docker Hub[6]. The app was developed specifically in context of aerOS Project.

Pilot1 - Business Process 4 - Activity - 1 (P1-BP4-DA5): Setting up POLIMI Side AGV Navigation System

As a consolidated software suite for research in robotics, POLIMI relied on ROS ecosystem7 to deploy its navigation system devoted to the Automated Guided Vehicle (AGV).

This software ecosystem has been chosen because of its structure (basically, it relies on pub-sub policy to allow different functional modules to exchange structured information), because of its language agnosticism (its modules are built via UNIX make utility tool8), and because of the availability of a large number of pre-built packages.

One of the most used modules for navigation of AGVs is indeed a pre-built package named "move_base", but, despite its robustness and its spread across research institutions9, its usage has been discarded. Indeed, move_base relies on two path planners10 as depicted above, a "global planner" which, via graph search algorithms, finds a sub-optimal path, and a "local planner", which, usually via Dynamic Window Approach 11simulation, enables dynamic avoidance of eventual obstacles in the environment.

[5] https://hub.docker.com/repository/docker/danny0117/aeros-dog2
[6] https://hub.docker.com/repository/docker/danny0117/aeros-persistor/general
[7] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In ICRA workshop on open-source software (Vol. 3, No. 3.2, p. 5).
[8] https://doi.org/10.1002/spe.4380090402
[9] https://doi.org/10.1016/j.promfg.2020.02.055
[10] https://doi.org/10.1109/SECON.2017.7925266
[11] https://doi.org/10.1109/100.580977

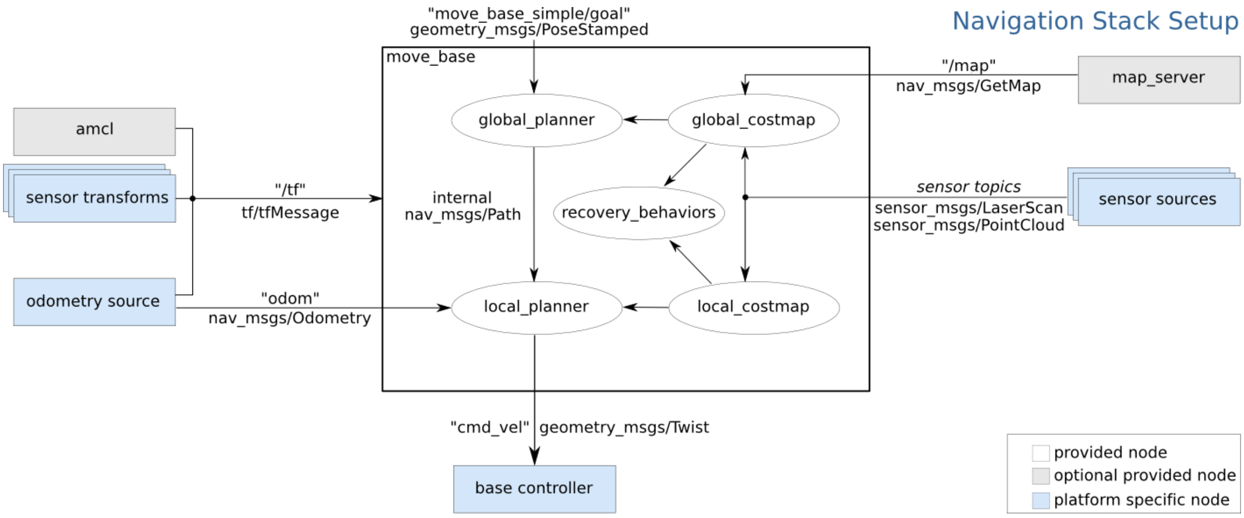


Figure 90. Pilot 1.4 Move base navigation stack

The reason for not relying on this tool sits indeed in the local planner and in the obstacle avoidance: counterintuitively, even if the obstacle avoidance feature enables a higher grade of resiliency, it has been demonstrated highly unsuitable for manned environments and to industrial shop floors in particular. This is due to the concept of perceived safety, which implies that workers feel safer whether the movements of robots are linear and replicated[12].

For this reason, POLIMI team decided to propose a navigation system closer to the industrial requirements. It relied on onboard libraries to implement linear movements on the horizontal or vertical direction in the environment map, and added on its top a simple segmentation mechanism to decompose the path from starting to ending point in sequential and orthographic movements, implemented into movements through the aforementioned libraries.

Pre-defined set of paths, listing all the possibile routes across POLIMI and MADE facilities, was then loaded into the mass memory of the AGV, limiting its dynamical flexibility, but enabling a higher level of reliability and predictability. The continuous localization of the AGV in the environment was provided thanks to Adaptive Monte Carlo Localization (AMCL) package[13].

From a safety-wise point of view, obstacle detection and collision avoidance were enabled via the laser-scanner mounted on the robot in such a way that, whenever a close-range obstacle was detected in the movement direction of the AGV, the vehicle just stopped until the obstacle (was) moved away from the path. Once again, this choice was made to increase the predictability of the AGV behavior, since other solutions, like move_base, present recovery routines that, when the AGV is stuck, make the vehicle spinning on its central axis, affecting the confidence of operators towards AGVs.

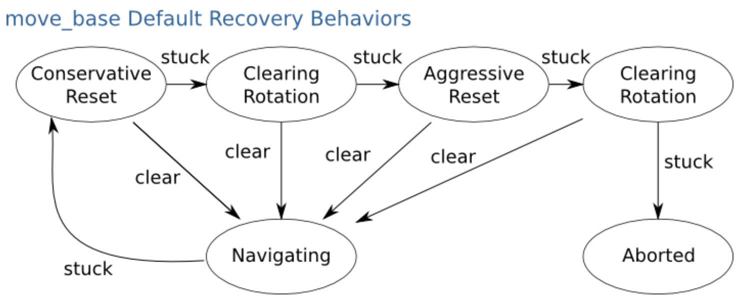


Figure 91. Pilot 1.4 Move base recovery behaviour

[12] https://doi.org/10.1007/s00170-023-11294-4
[13] https://doi.org/10.23919/ChiCC.2018.8482698

During the prototyping phase, and available on-site for continuous monitoring, the ROS-native tool RViz was deployed.

From Figure 92, it can be seen the window-wise division of its GUI, where in top-left frame are depicted the active ROS topics, in the bottom-left frame the onboard camera is streaming the AGV's point of view, while in the big right frame the localization of the AGV is represented in current time with respect to the map of the environment (in this case, MADE floor). Worth to be noticed is the possibility to intuitively set a target point via the 2D Nav Goal button. This function bypasses the developed path planner and enables a full control from the research team, allowing to manage unpredictable circumstances.

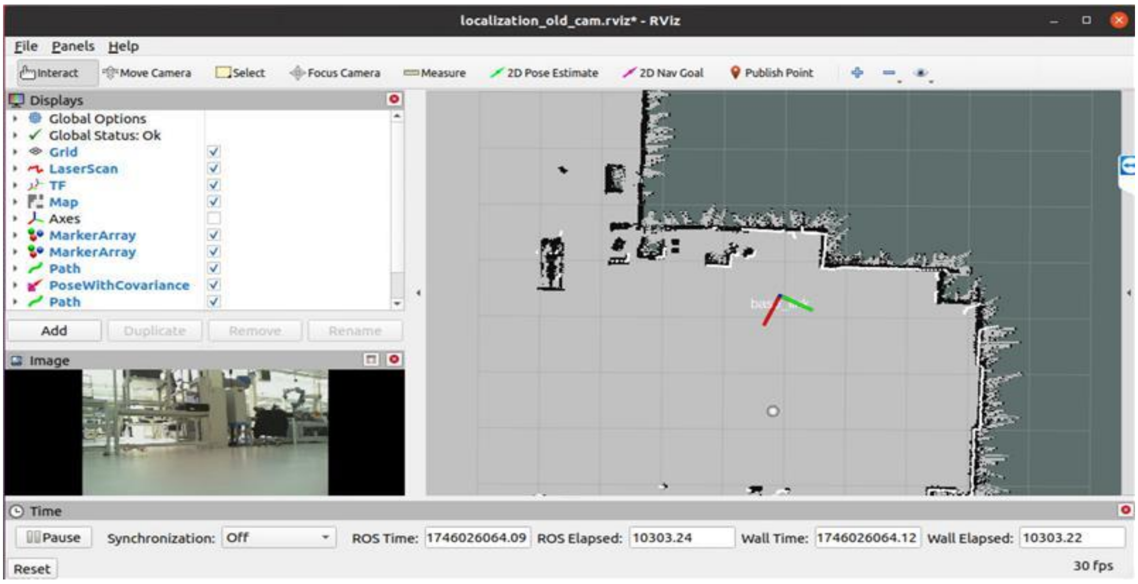


Figure 92. Pilot 1.4 Setting up POIMI Side AGV Nagivation System

Data exchange between the AGV and higher layers of the aerOS architecture was ensured by tailor-made parsers which converted the JSON data structure of ROS messages into the one required by NGSI-LD structure, coupled with communication interfaces advertising/consuming the NGSI-LD structured JSON data via simple REST calls.

Pilot1 - Business Process 4 - Activity - 1 (P1-BP4-DA6): Development of Warehouse commander for AGV-POLIMI

This application is used for commanding the AGV to supply materials from the warehouse to the production station in the POLIMI site. The app has a simple web interface-based button which the operator presses when the materials have been loaded on the AGV and ready to be sent to the workstation for production. Currently this is deployed on the Raspberry pi 5 in the POLIMI domain. In the pilot scenario this app is run from a mobile device carried by the warehouse operator.

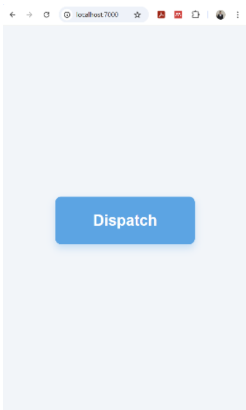


Figure 93. Pilot 1.4 AGV Commander App

The app was developed in Python. Considering the fact that the Pilot uses both ARM 64 and AMD64 IE elements, A dual Architecture (ARM64 and AMD64) image of the app was developed and is openly available on Docker Hub[14]. The app was developed specifically in context of aerOS project.

Pilot1 - Business Process 4 - Activity - 1 (P1-BP4-DA7): Development of Communication Api between AGV Nav. And POLIMI Front end app

For the AGV to communicate directly with the front-end, a Flask server and REST API are used. The front-end app can send orders directly to the AGV, and Flask server interprets these orders, converting them into ROS topics for path planning and navigation. As shown below, illustrating the AGV command center output: on the right, the AGV server is running as a ROS node; at the top left, we display the ROS topic updates for order status; and at the bottom, the navigation ROS node output after receiving a destination command.

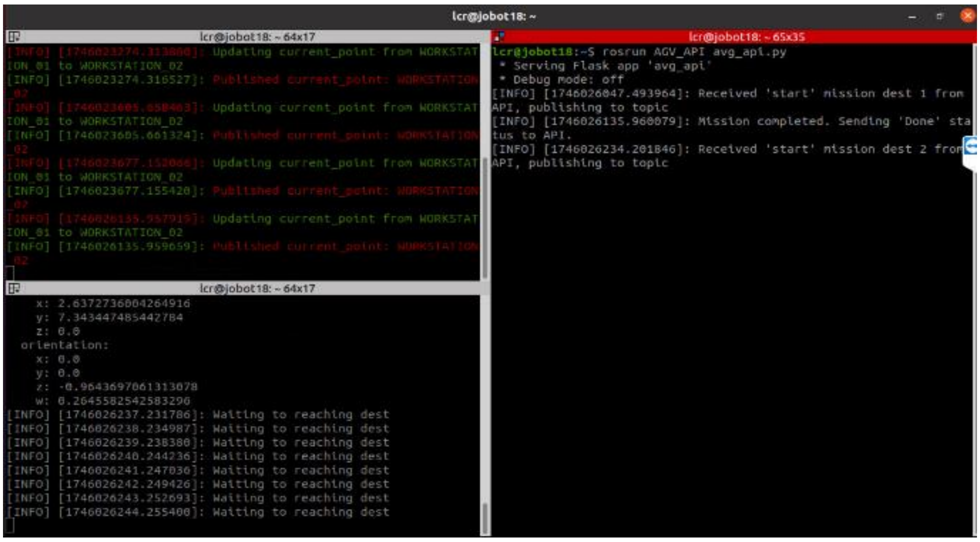


Figure 94. Pilot 1.4 Communication API between AGV Navigation and POLIMI front-end app

Pilot1 - Business Process 4 - Activity - 1 (P1-BP4-DA8): Development of AI model to optimize the number of valves

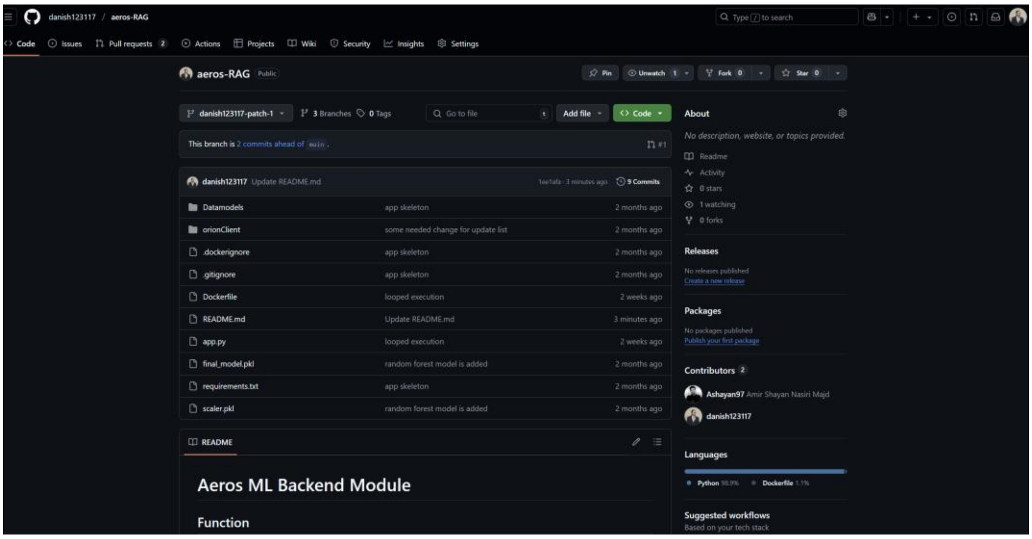


Figure 95. Pilot 1.4 Development of AI model

The optimizer app is used for automatic order outsourcing of orders to the POLIMI which acts as an external factory in this scenario. The outsourcing decision is based on the parameters like existing queue size, moving

[14] https://hub.docker.com/repository/docker/danny0117/aeros-agvbutton/general

average interarrival time of orders, line engagement status, line breakdown status and time to complete current order. A random forest ML model is used for the decision and it runs every 5 minutes. The app is runs completely on the backend and the Git repository [15]is available. The app was developed specifically in context of aerOS Project.

2.1.4.2.1. Integration activities

Pilot1 – Business Process 4 – Activity - 1 (P1-BP4-IA1): aerOS Basic components

Done for all domains, for MADE domain some trouble shooting required for deploying applications from Management portal.

```
local_admin@servaep4ma01:~$ kubectl get pods
NAME                                          READY   STATUS    RESTARTS       AGE
aeros-k8s-shim-86dfd587bf-mpr94               1/1     Running   0              17d
api-gateway-krakend-8566f944b7-6fdfn          1/1     Running   0              16d
federator-7f45fc4595-m5wnv                    1/1     Running   0              17d
hlo-allocator-fdf5bf7bd-58v57                 1/1     Running   0              15d
hlo-data-aggregator-5f5cff48d-zb487           1/1     Running   0              15d
hlo-deployment-engine-6795bd95bf-rswpz        1/1     Running   0              15d
hlo-frontend-774c96f68d-dnwfd                 1/1     Running   0              15d
idm-database-0                                1/1     Running   0              143d
idm-keycloak-6c68867575-ch88w                 1/1     Running   0              42d
iota-api-7857f54798-jqkvt                     1/1     Running   0              140d
iota-coordinator-57c5469f98-wrdl9             1/1     Running   0              140d
iota-dashboard-58fc54948d-c8nml               1/1     Running   0              140d
iota-hornet-5db6db94b-q8r5k                   1/1     Running   0              140d
iota-hornet-8szb9                             2/2     Running   0              140d
iota-hornet-cgtm6                             2/2     Running   0              140d
llo-k8s-controllermanager-56dcff98c6-w8dww    2/2     Running   0              16d
management-portal-backend-854d687675-z92t6    1/1     Running   0              14d
management-portal-frontend-7c7757b96-lqsxw    1/1     Running   0              14d
openldap-0                                    1/1     Running   0              143d
openldap-ltb-passwd-9f54768b6-cc2zj           1/1     Running   73 (33d ago)   143d
openldap-phpldapadmin-685fc7c86d-k57xj        1/1     Running   63 (33d ago)   143d
orion-ld-broker-5bf4f68b4f-9s6vx              1/1     Running   0              16d
orion-ld-mongodb-0                            1/1     Running   0              16d
self-awareness-hardwareinfo-7vd6d             1/1     Running   0              80d
self-awareness-hardwareinfo-m7qrc             1/1     Running   0              80d
self-awareness-hardwareinfo-tsn8g             1/1     Running   0              80d
self-awareness-powerconsumptionamd64-744bt    1/1     Running   0              80d
self-awareness-powerconsumptionamd64-bngxn    1/1     Running   0              80d
self-awareness-powerconsumptionamd64-xd65t    1/1     Running   0              80d
self-orchestrator-orchestrator-4nzvn          1/1     Running   0              140d
self-orchestrator-orchestrator-d4p8c          1/1     Running   0              140d
self-orchestrator-orchestrator-dnr78          1/1     Running   0              140d
```

Figure 96. Pilot 1.4 NASERTIC Domain – aerOS basic components installation

In the figure above we can see the aerOS services deployed on the NASERTIC domain. Since the NASERTIC domain consists of a k8 cluster with a master node and two worker nodes, only single snapshot can be seen.

[15] https://github.com/danish123117/aeros-RAG/tree/danish123117-patch-1

```
aeros_polimi@aerospolimi:~$ kubectl get pods
NAME                                         READY   STATUS    RESTARTS          AGE
aeros-k8s-shim-6dc97f99cf-wf6h7              1/1     Running   0                 17d
api-gateway-krakend-b4df8899f-6bls7          1/1     Running   2 (17d ago)       50d
federator-5b57ff65df-4vkjr                   1/1     Running   1 (16d ago)       16d
hlo-allocator-8478b9d46-xnxw7                1/1     Running   0                 15d
hlo-data-aggregator-7d89c786bb-vp9qf         1/1     Running   0                 15d
hlo-deployment-engine-8698885479-rr9t8       1/1     Running   0                 15d
hlo-frontend-c9bfc8fb6-ngsqj                 1/1     Running   0                 15d
iota-api-7b7647c5c9-k8275                    1/1     Running   4 (17d ago)       140d
iota-dashboard-69ff5c4ccd-r9wsp              1/1     Running   113 (5h7m ago)    16d
iota-hornet-qxllp                            2/2     Running   0                 16d
llo-k8s-controllermanager-84b89b66f6-46dll   2/2     Running   158 (13m ago)     14d
orion-ld-broker-64fb4bb8db-pqqjf             1/1     Running   0                 16d
orion-ld-mongodb-0                           1/1     Running   0                 16d
self-awareness-hardwareinfo-tnkz2            1/1     Running   0                 16d
self-awareness-powerconsumptionamd64-xbwjk   1/1     Running   0                 16d
self-orchestrator-orchestrator-q5rnf         1/1     Running   4 (17d ago)       140d
aeros@aeros-desktop:~/Desktop/aeros/self-orchestration$ sudo docker container ls
CONTAINER ID   IMAGE                                                                                       COMMAND               CREATED          STATUS               PORTS
                 NAMES
7542450f3424   registry.gitlab.aeros-project.eu/aeros-public/common-deployments/self-orchestrator:1.2.0                 "node script.js"      12 seconds ago   Up 12 seconds        0.0.0.0:8001
->8001/tcp, :::8001->8001/tcp   aeros_self-orchestrator_ctr
6eb9324bf1c3   registry.gitlab.aeros-project.eu/aeros-public/common-deployments/self-awareness/hardware_info:1.4.0   "python3 script.py"   2 weeks ago      Up About a minute
                 aeros_self-awareness_hardware-info_ctr
lcr@jobot18:~/Desktop/aros/selforc$ sudo docker container ls
CONTAINER ID   IMAGE                                                                                       COMMAND               CREATED
                 NAMES
9fd1cf8c0d2a   registry.gitlab.aeros-project.eu/aeros-public/common-deployments/self-awareness/hardware_info:1.4.0   "python3 script.py"   2 weeks ago
                 aeros_self-awareness_hardware-info_ctr
fccadbb2427b   registry.gitlab.aeros-project.eu/aeros-public/common-deployments/self-orchestrator:1.2.0                 "node script.js"      2 weeks ago
tcp, :::8001->8001/tcp   aeros_self-orchestrator_ctr
lcr@jobot18:~/Desktop/aros/selforc$
```

Figure 97. Pilot 1.4 POLIMI Domain – aerOS basic components installation

Figure 97 displays a similar snapshot of the POLIMI domain. However here there are 3 separate sub snapshots. The reason for this is that here there is a single node k3s cluster hosted on an Industrial PC and 2 IEs which are docker based (AGV and the Raspberry pi 5) and is possible to see on these 2 IEs there are only have the self* tools installed.

```
MADESUC0902@aeros:~$ sudo kubectl get pods
NAME                                         READY   STATUS             RESTARTS         AGE
aeros-k8s-shim-6dc97f99cf-w9cfr              1/1     Running            1 (7d1h ago)     12d
api-gateway-krakend-b4df8899f-mh49f          1/1     Running            1 (7d1h ago)     12d
federator-846c6b5f7d-1q79v                   1/1     Running            3 (7d1h ago)     12d
hlo-allocator-68ff66bfb8-b6gkr               1/1     Running            1 (7d1h ago)     12d
hlo-data-aggregator-7d89c786bb-r8k2t         1/1     Running            1 (7d1h ago)     12d
hlo-deployment-engine-8698885479-wnvrr       1/1     Running            1 (7d1h ago)     12d
hlo-frontend-c9bfc8fb6-fsb5j                 1/1     Running            1 (7d1h ago)     12d
iota-api-7b7647c5c9-vk4sr                    1/1     Running            3 (7d1h ago)     159d
iota-coordinator-78c96f977d-9fzck            0/1     CrashLoopBackOff   10882 (2m47s ago)   159d
iota-dashboard-cd588bdf8-rng6x               0/1     CrashLoopBackOff   10882 (57s ago)  159d
iota-hornet-64548957f6-bdtzh                 0/1     CrashLoopBackOff   11909 (95s ago)  159d
llo-k8s-controllermanager-84b89b66f6-mhlnp   2/2     Running            3 (7d1h ago)     12d
nginx-7769f8f85b-2dndn                       1/1     Running            1 (7d1h ago)     12d
orion-ld-broker-75db9dccc9-z4srv             1/1     Running            1 (7d1h ago)     12d
orion-ld-mongodb-0                           1/1     Running            1 (7d1h ago)     12d
self-awareness-hardwareinfo-8hf4s            1/1     Running            2 (7d1h ago)     12d
self-awareness-powerconsumptionamd64-jk5t8   1/1     Running            2 (7d1h ago)     12d
self-orchestrator-orchestrator-975wv         1/1     Running            1 (7d1h ago)     12d
```

```
aeros-made@aerosmade:~$ sudo docker container ls
[sudo] password for aeros-made:
CONTAINER ID   IMAGE                                                                                       COMMAND               CREATED       STATUS       PORTS
  NAMES
35e19f855d07   registry.gitlab.aeros-project.eu/aeros-public/common-deployments/self-orchestrator:1.2.0                 "node script.js"      12 days ago   Up 12 days   0.0.0.0:8001->8001/tcp, :::8001->8001/tcp
  aeros_self-orchestrator_ctr
7ef08661f713   registry.gitlab.aeros-project.eu/aeros-public/common-deployments/self-awareness/hardware_info:1.4.0   "python3 script.py"   12 days ago   Up 12 days
  aeros_self-awareness_hardware-info_ctr
```

Figure 98. Pilot 1.4 MADE Domain – aerOS basic components installation

Similar to the POLIMI domain, MADE domain (see Figure 98) has one single node k3s cluster hosted on an existing server VM, and a raspberry pi 5 which is running docker. Therefore, we can see 2 sub snapshots here

This is consistent with the overall aerOS continuum architecture of Pilot 1.4 as already displayed in Figure 81.

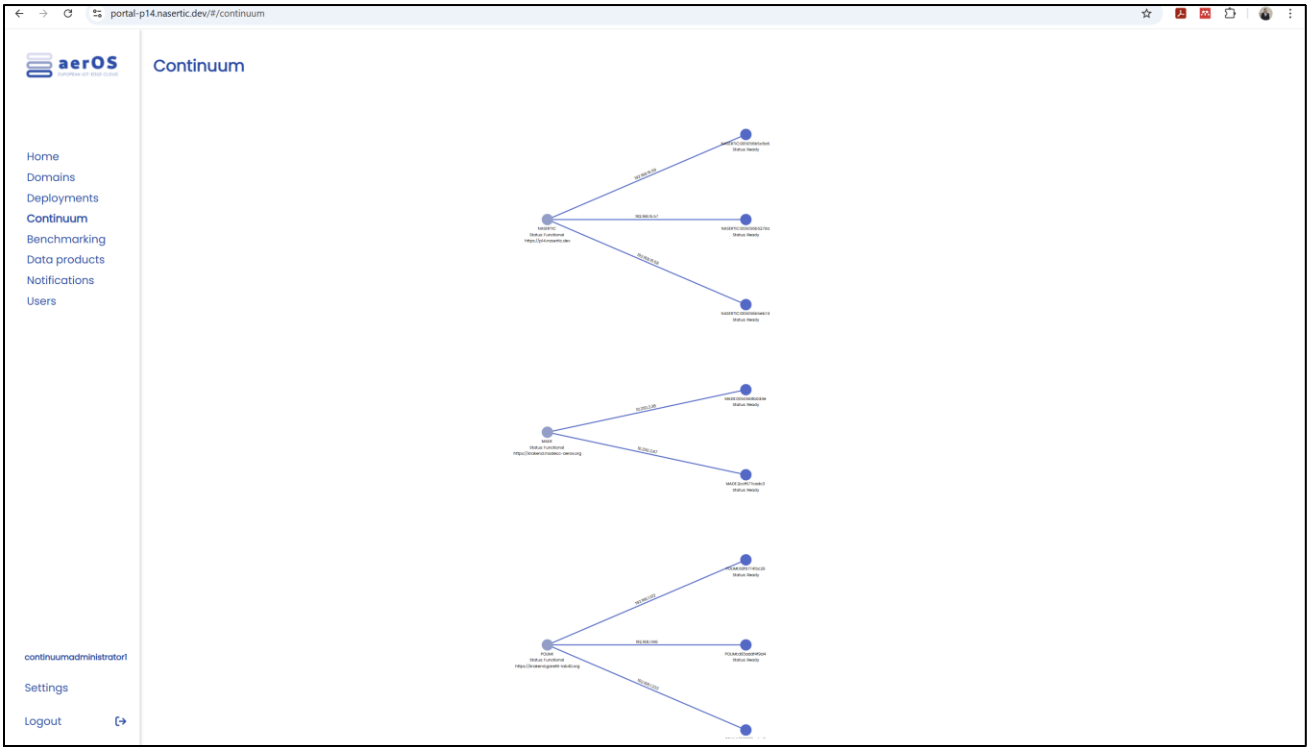


Figure 99. Pilot 1.4 Management portal – aerOS continuum

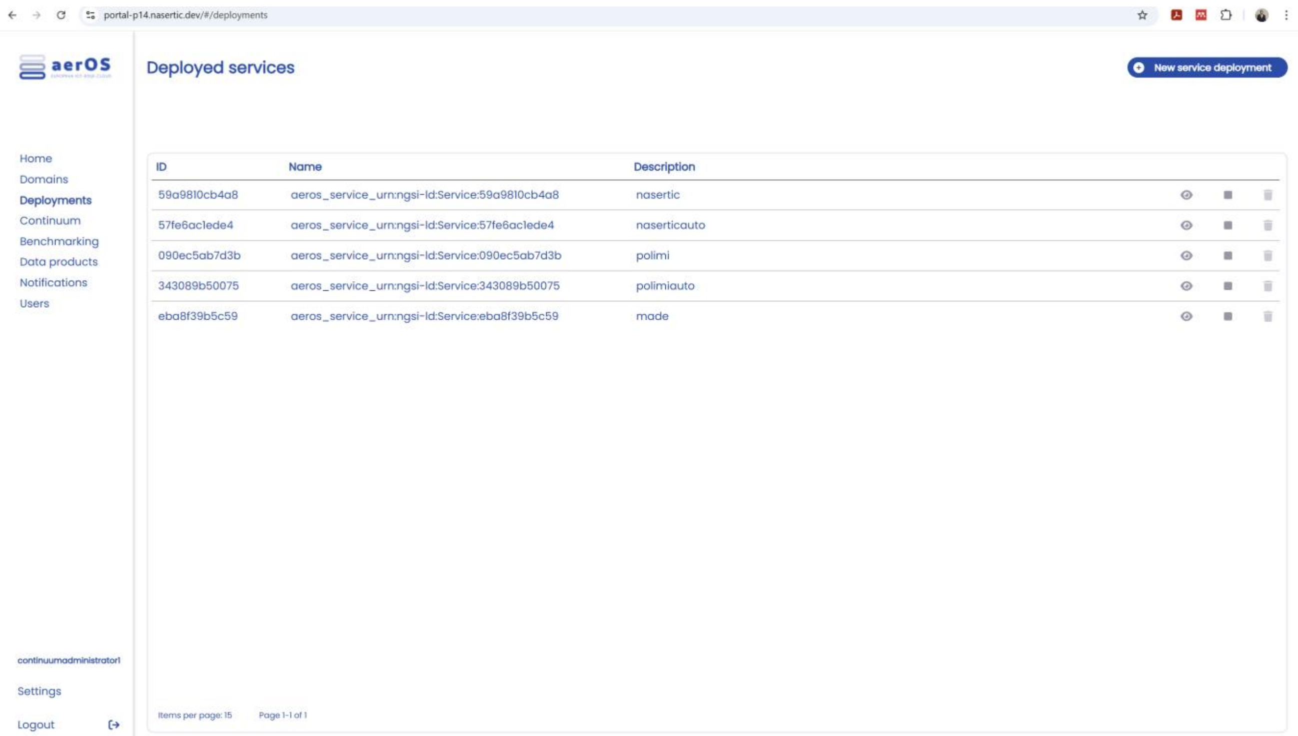


Figure 100. Pilot 1.4 Management portal – Deployed services

The Figures above shows a couple of snapshots from the management portal showing the continuum with all the infrastructure elements from all 3 domains, additionally we can also see some deployments that were made in different domains of the Pilot 1.4 aerOS continuum.

Pilot1 – Business Process 4 – Activity - 2 (P1-BP4-IA2): aerOS Non-Basic components

During the last project period, testing of the application of frugality, as provided by project partners, has been initiated. Different approaches were explored and currently on testing to the developed AI model. September 2025 will be dedicated to evaluating the results and determining the applicability of the most effective frugality approach. The last two project months will be used for implementing and testing the Embedded Analytics Tool (EAT) within the application.

Pilot1 – Business Process 4 – Activity - 3 (P1-BP4-IA3): Integrate Applications with Orion Context broker

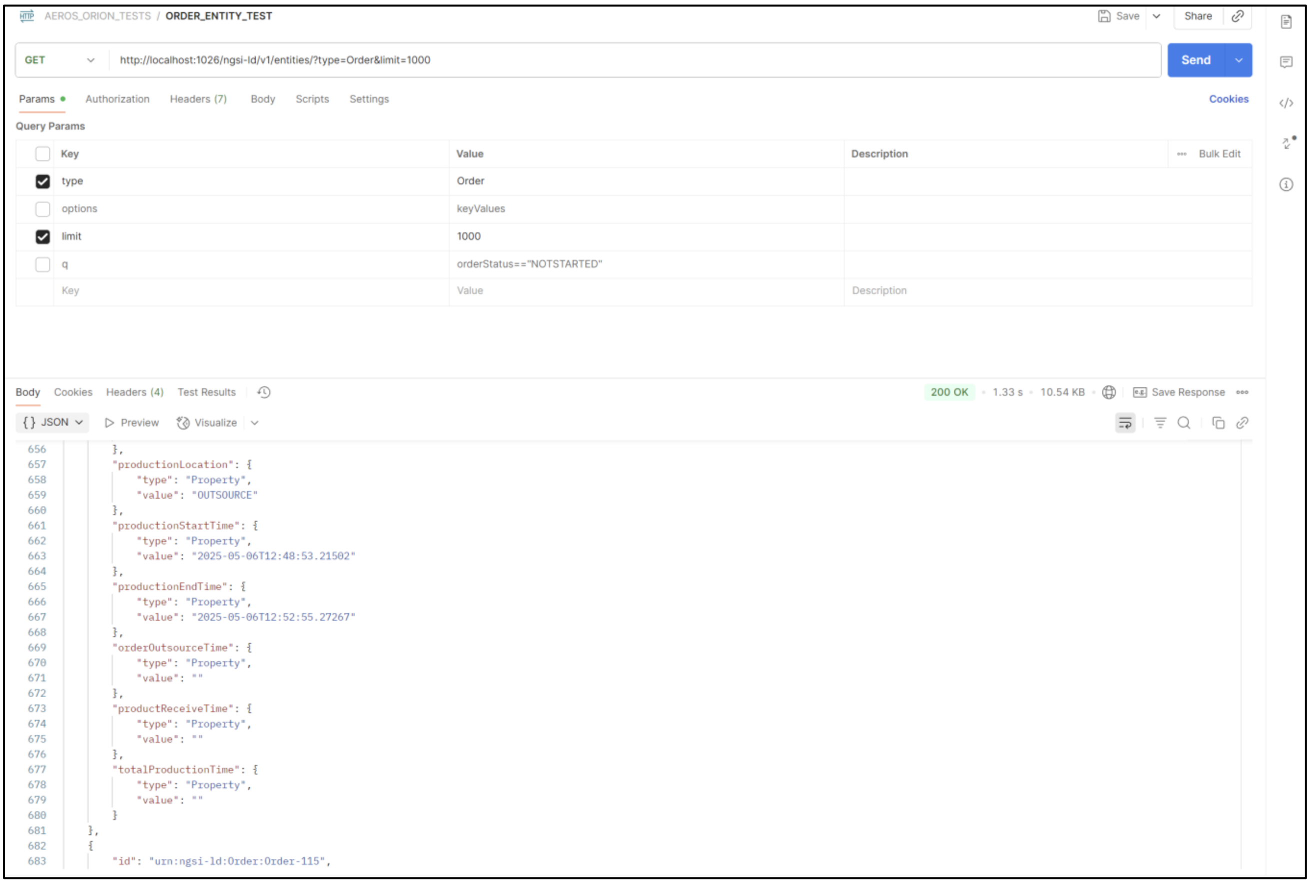


*Figure **101**. Pilot 1.4 Snapshot of interacting with aerOS Data Fabric (Orion-LD request)*

Figure above shows a snapshot of a GET request to Orion-LD showing the order entities stored inside the Orion-LD (essential element of aerOS' Data Fabric).

Pilot1 – Business Process 4 – Activity - 4 (P1-BP4-IA4): APIs acces provision for existing Order/infrastructure management for MADE

Several modifications were implemented on the LEA order launch platform. Specifically, REST API calls were exposed to enable the indirect initiation of production orders and to retrieve output data from the process.

To support order initiation, a POST call was developed, accepting the following parameters: the name of the product to be manufactured, the product type, the execution mode of the process, and the quantity to be produced.

Additionally, GET calls were implemented to allow querying of the LEA platform regarding the progress status of ongoing production orders, as well as the completion or cancellation status of past orders.

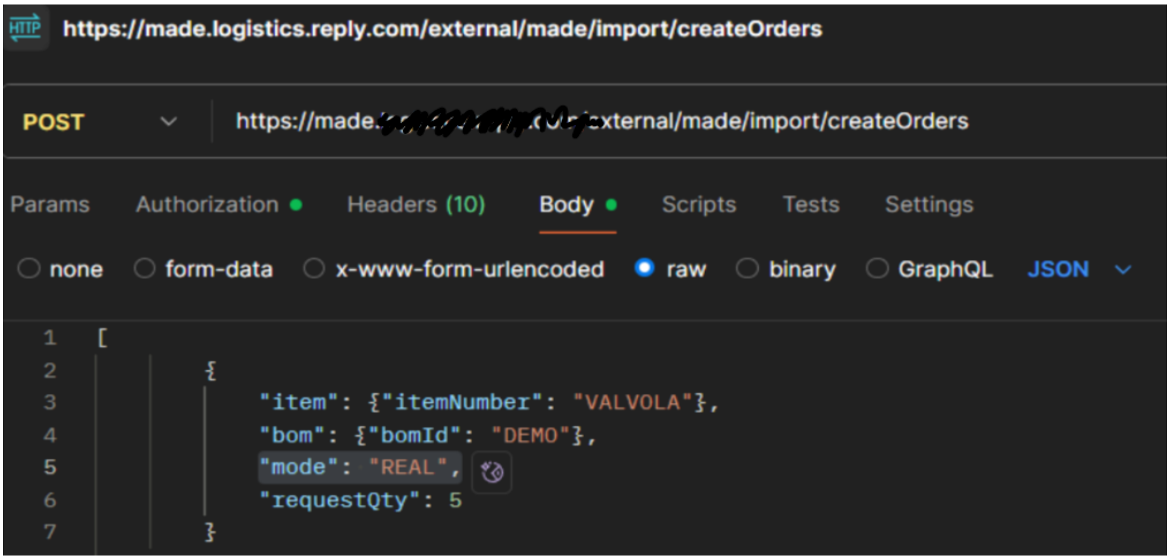


*Figure **102**. Example of an interaction (POST request) with LEA platform (MADE)*

2.1.5. Pilot 1 Time-plan

The following Gantt diagram list all the activities from M19 (February 2024) until M35 (July 2025), representing evolution from status declared in D5.3.

Table 5. Pilot 1 updated Gantt timeline

PILOT 1		2024											2025								
Code	**Name**	**M19**	**M20**	**M21**	**M22**	**M23**	**M24**	**M25**	**M26**	**M27**	**M28**	**M29**	**M30**	**M31**	**M32**	**M33**	**M34**	**M35**	**M36**	**M37**	**M38**
Pilot 1.1 - (Business Process 1) - Green Manufacturing and CO2 Footprint Monitoring																					
Setup & Procurement Activities																					
P1-BP1-SA1	Stations identification for the trial																				
P1-BP1-SA2	Hardware setup																				
P1-BP1-SA3	Equipment configuration																				
Development Activities																					
P1-BP1-DA4	Definition of IT architecture																				
P1-BP1-DA5	Communication infrastructure developed or adapted																				
P1-BP1-DA6	APIs setup																				
P1-BP1-DA7	Software configuration / development																				
Integration Activities																					
P1-BP1-IA8	Identification of place of deployment in current architecture																				
P1-BP1-IA9	Requirements definition for deployment																				
P1-BP1-IA10	Test energy data collection																				
P1-BP1-IA11	Adjust equipment configuration according to first feedback																				
P1-BP1-IA12	aerOS Basic components																				
P1-BP1-IA13	aerOS Non Basic components																				
P1-BP1-IA14	Dashboard creation with energy data																				
P1-BP1-IA15	Integration of data analysis service for reports and statistics creation																				
P1-BP1-IA16	Creation of a future prediction of the CO2 footprint for each product																				
P1-BP1-IA17	Experimenting with Gaia-X services																				
Validation Activities																					
P1-BP1-VA18	Data quality verification																				
P1-BP1-VA19	Digital Product Passport implementation																				
P1-BP1-	Improvement activities																				

VA20																					
P1-BP1-VA21	KPIs validation																				
P1-BP1-VA22	Qualitative validation																				
P1-BP1-VA23	Evaluation and reporting																				
Pilot 1.2 - (Business Process 2) - Automotive Smart factory Zero defect manufacturing																					
Setup & Procurement Activities																					
P1-BP2-SA1	Installation of control camera in the CMM																				
P1-BP2-SA2	Installation of pressure sensor in the CMM																				
Development Activities																					
P1-BP2-DA5	Enable Software-based control services and IoT hub for collection and brokering for instrumentation information																				
Integration Activities																					
P1-BP2-IA6	aerOS Components																				
Validation Activities																					
P1-BP2-VA7	Remote configuration/set-up of the CMM instrumentation robotic and kinematic configuration																				
P1-BP2-VA8	Remote tactile operation of CMM																				
P1-BP2-VA9	aerOS assist and optimize the process of Digital Twin creation																				
P1-BP2-VA10	Dynamic execution of metrology services and Data assembling																				
P1-BP2-VA11	KPI 2.1.1: Production process accuracy >10% baseline																				
P1-BP2-VA12	KPI 2.1.2: Digital service programming time 2 days																				
P1-BP2-VA13	KPI 2.1.3: Dimensional quality control productivity 5parts/hr																				
Pilot 1.3 - (Business Process 3) - Zero Ramp-up safe PLC reconfiguration for Lot-Size-1 Production																					
Setup & Procurement Activities																					
P1-BP3-SA3	Procurement and setup of 2 robot arm modules																				
P1-BP3-SA5	Setup Tech Hall Network																				
P1-BP3-SA6	Procurement camera for robot arm																				
P1-BP3-SA7	Setup AI camera detection																				
P1-BP3-SA8	Procurement and setup gripper for robot arm																				

Development Activities																					
P1-BP3-DA2	AGV Fleet Manager																				
P1-BP3-DA3	Device simplified programming over low code tools																				
P1-BP3-DA5	Siemens Industrial Edge configuration																				
P1-BP3-DA6	Integration of TSN in a domain																				
Integration Activities																					
P1-BP3-IA2	Custom LLO for Siemens Industrial Edge device																				
P1-BP3-IA3	Communicate domains using open ziti																				
P1-BP3-IA4	aerOS Basic components																				
P1-BP3-IA5	aerOS Non Basic components																				
P1-BP3-IA6	Definition of data federation – and achieving such federation																				
Validation Activities																					
P1-BP3-VA1	KPI 2.1.7 validation: AGV usage > 80 %																				
P1-BP3-VA2	KPI 2.1.8 validation: AGV availability > 95%																				
Pilot 1.4 - (Business Process 4) - AGV Swarm Zero break-down logistics for Lot-Size-1 Production																					
Setup & Procurement Activities																					
P1-BP4-SA1	Upgrade/Update POLIMI AGV																				
P1-BP4-SA2	Definition of IT architecture with aerOS																				
P1-BP4-SA3	Procurement and setup of Raspberry Pi POLIMI																				
P1-BP4-SA4	Procurement and setup of Raspberry Pi MADE																				
P1-BP4-SA5	Setup of Domain Infrastructure																				
P1-BP4-SA6	Setup of Dedicated network for MADE Domain																				
Development Activities																					
P1-BP4-DI1	Development/containerisation of overlay Order management application at MADE																				
P1-BP4-DI2	Development/containerisation of overlay order management at POLIMI																				
P1-BP4-DI3	Development of Synthetic order generator application																				
P1-BP4-DI4	Development of order backend persistor																				
P1-BP4-DI5	Setting up POLIMI Side AGV Navigation System																				
P1-BP4-DI6	Development of AGV POLIMI side front end																				

P1-BP4-DI7	Development of Communication Api between AGV Nav. And POLIMI Front end app																				
P1-BP4-DI8	Development of AI model to optimize the number of valves																				
Integration Activities																					
P1-BP4-IA4	aerOS Basic components																				
P1-BP4-IA5	aerOS Non Basic components																				
P1-BP4-IA6	Integrate Applications With Orion Context broker																				
P1-BP4-IA7	APIs acces provision for existing Order/infrastructure management for MADE																				
Validation Activities																					
P1-BP4-VA1	Validation of KPI 2.1.9																				
P1-BP4-VA2	Validation of KPI 2.1.10																				

2.2. Pilot 2 - Containerised Edge Computing near Renewable Energy Sources

Pilot 2 of aerOS **revolves around shifting computation of workloads into edge nodes and optimizing the use of renewable (photovoltaic) energy source, in changing green-energy availability conditions**.

At the beginning of the WP5, the physical deployment environment was designed, and suitable site was selected. Activities focused on obtaining hardware: servers, switches, and RACKs, and assembling them into two edge nodes. For this pilot, an edge node is a small or medium-sized collection of servers, enclosed in a metal container with appropriate internal networking and air conditioning hardware, as well as attachment points for external hardware. The metal container allows for quick deployment of a "server room" edge node. Installation of the hardware and running containerized tests of aerOS on the edge, as well as in the lab environment, were also completed as part of initial activities. A node was initially deployed in a test location, before being moved to the final deployment spot.

Later, during software development activities, aside from deployment of aerOS components, workloads were prepared to ensure, that the task that was previously computed in a centralized cloud, could be completed in independent edge nodes. The completed preparations also included addressing security concerns. Required modifications of aerOS orchestration systems were developed, alongside custom components, to allow for automatic orchestration of independent, or serial jobs, that load and unload onto and from an edge node.

The hardware and software setup for containerized edge computing is validated by letting aerOS orchestrate the prepared workloads, that are run in the edge nodes, connected to a green energy source. A node is also connected to and partially powered by an additional set of photovoltaic panels, that deliver variable extra power, depending on weather conditions. An energy availability prediction service is integrated with the edge node, as well as a nearby weather station, to deliver predicted green power level to the system.

Model for real-time collection and forecasting of electricity prices on the Polish energy market (TGE) was developed. The solution includes containerized services for data scraping and LSTM-based prediction, integrated with monitoring dashboards. Ongoing activities focus on implementing continuous model evaluation, drift detection, and periodic fine-tuning.

Building on this foundation, identical edge nodes can be deployed to other European energy exchanges, each will locally ingest and retrain on its national market data before sharing only neural-network weight updates with a central aggregator. **This federated learning setup preserves data sovereignty and minimizes network traffic, perfectly embodying our Frugal AI goal of efficient, privacy-preserving model improvement**.

2.2.1. Green Edge Processing

Scenario “Green Edge Processing” aims to deploy two federated edge nodes and a private CF cloud located directly at renewable energy premises, and connected to different smart devices and data sources from PhV source operated by CF.

Since the system must be managed in an energy-efficient, network-aware, and self-conscious manner, aerOS has proven ideal for measuring the reductions achieved (benchmarking parameters based on real-time own analytics in the IE) in the orchestration of tasks deployed on the edge rather than in the cloud (e.g., AI). As computing resource requirements, available energy, or network throughput change, aerOS will facilitate rapid changes (self-scalability, self-automation) in task distribution through orchestration (managing topology, tasks, and services).

The following table showcases how our pilot has switched from processing data in the Cloud to the Edge.

Table 6. Pilot 2 Edge vs Cloud comparison before and after aerOS.

	Cloud	Edge
Before aerOS	Earth Observation products processing	None
After aerOS	Management and High-Level Orchestration (HLO) of edge nodes	Earth Observation products processing Custom workloads processing (different kinds of workloads depending on use case)

2.2.1.1. Technical schema

The Pilot 2 Edge-to-cloud continuum consists of four domains:

1. One entry point domain in the CloudFerro Cloud
2. Two regular domains located in containerized edge-nodes
3. Electrum domain (added on April 2025)

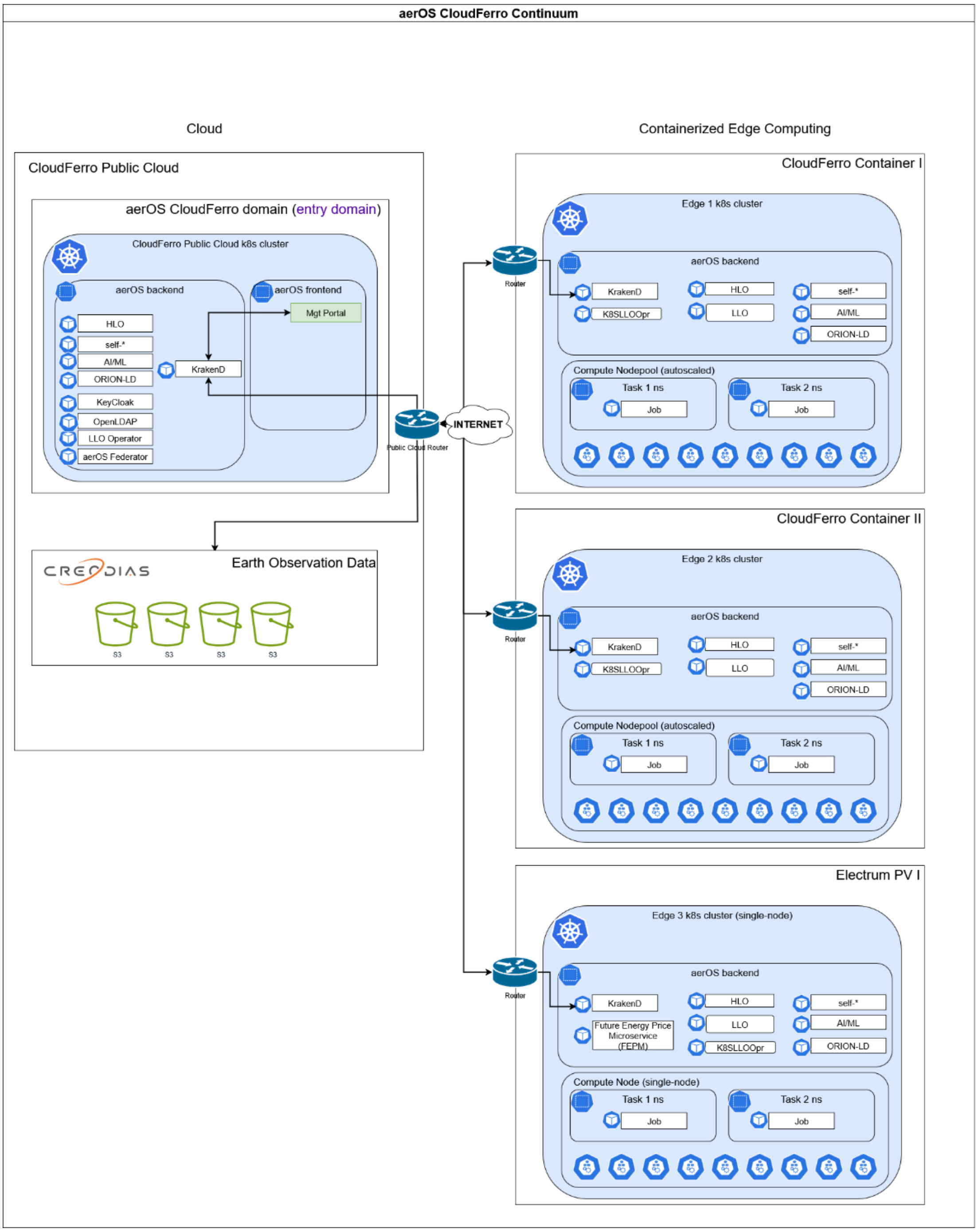


Figure 103. Pilot 2 Topology diagram – Technical Schema

The adopted architecture for the first scenario—Green Edge processing—is the schema that can be seen in the next figure.

It covers Earth Observation data processing such as cloud mask computing or similar jobs good for computing on CPU hardware. User launches one job per each node and that job gets item's data for processing from central queue until all done. The example of Scenario 1 processing is to calculate Cloud Mask of Sentinel-2 imaging of at least Poland area for 2024 with SEnSeIv2 [16]library.

[16] https://github.com/aliFrancis/SEnSeIv2

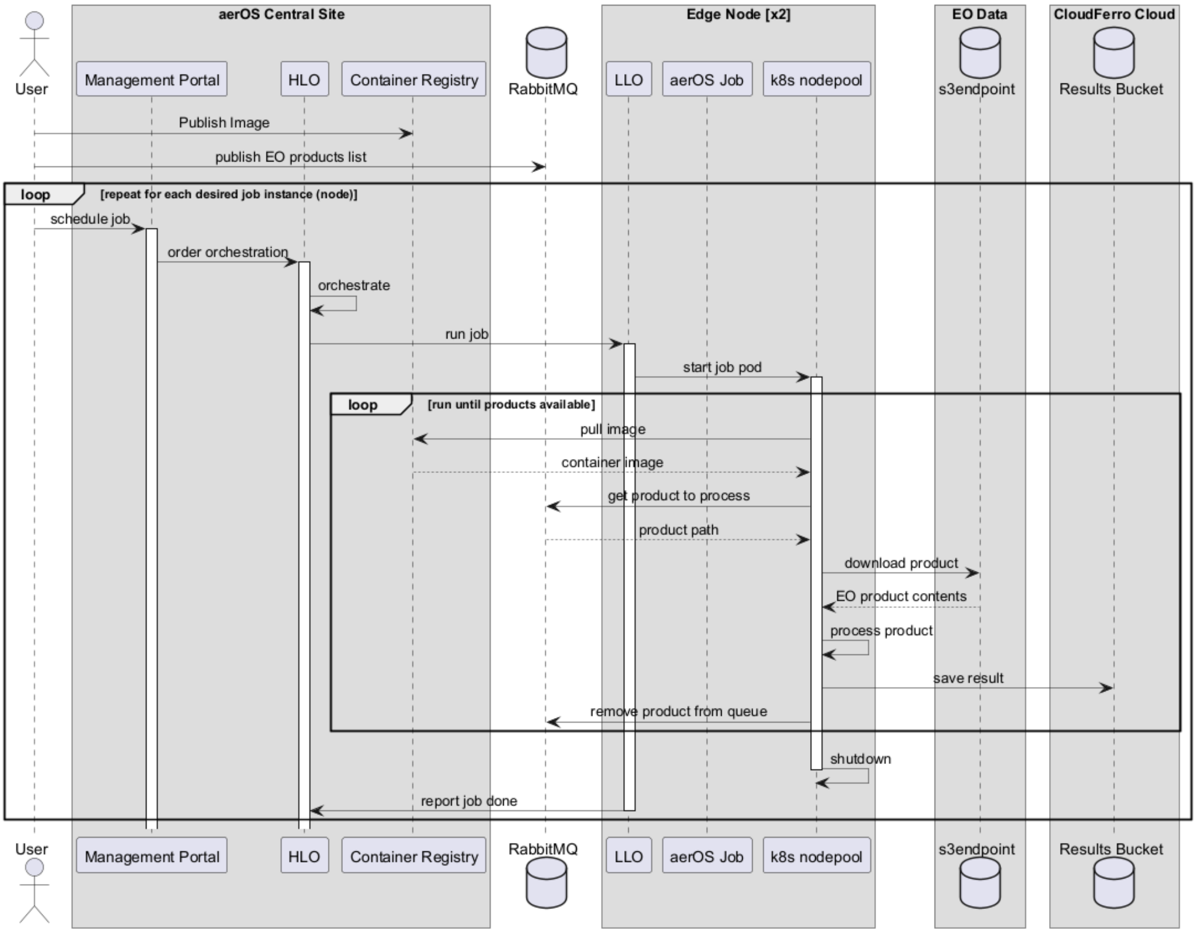


Figure 104. Pilot 2 Green Edge Processing Ontology diagram

Depending on the task, CPU is still good enough to calculate Cloud Mask. GPU is much faster however not cost effective.

There is the need of adding the time to download the satellite product from the repository, due to its size (~1GB) there is no point investing in GPU while very cheap CPU servers are available.

2.2.1.1. Report of activities

2.2.1.1.1. Setup and procurement activities

The servers and switches were obtained in M19 (March 2024), the racks, servers and switches for the second edge node (container) have been obtained in month M22 (June 2024).

Provided environment consists of two edge nodes, each one containing:

- Two server racks
- One router node with following specification: 2 processors 12 core 2.3 GHz, 128 GB RAM, 2 x SSD
- Boot Disk 1TB, ethernet port 4xSFP+ 25G, 4xSFP+ 10G, 2xUTP 1G, IPMI, redundant power supply
- One control plane node with following specification: 2 processors 12 core 2.3 GHz (Intel Xeon E5-2670 v3), 256 GB RAM, 2 x SSD Boot Disk 960 GB each, ethernet port 2xSFP+ 25G, 2xUTP 1G, IPMI, redundant power supply
- 37 compute nodes with following specification: 2 processors 12 core 2.3 GHz (Intel Xeon E5-2670 v3), 256 GB RAM, 2 x SSD Boot Disk 120GB, IPMI, redundant power supply
- 3 storage nodes with following specification: 2 processors 12 core 2.3 GHz (Intel Xeon E5-2670 v3), 256 GB RAM, 2 x SSD Boot Disk 240GB, ethernet port 4xSFP+ 25G, 2xUTP 1G, 22*0,96TB SSD, IPMI, redundant power supply.

2.2.1.1.2. Development activities

Pilot2 - Business Process 1 - Activity - 4 (P2-BP1-DA4): HW installation and run test in container

The containers were installed on CF site; the first batch of hardware was mounted and connected inside and the cooling system 'door cooler' was tested, as presented in the next image.

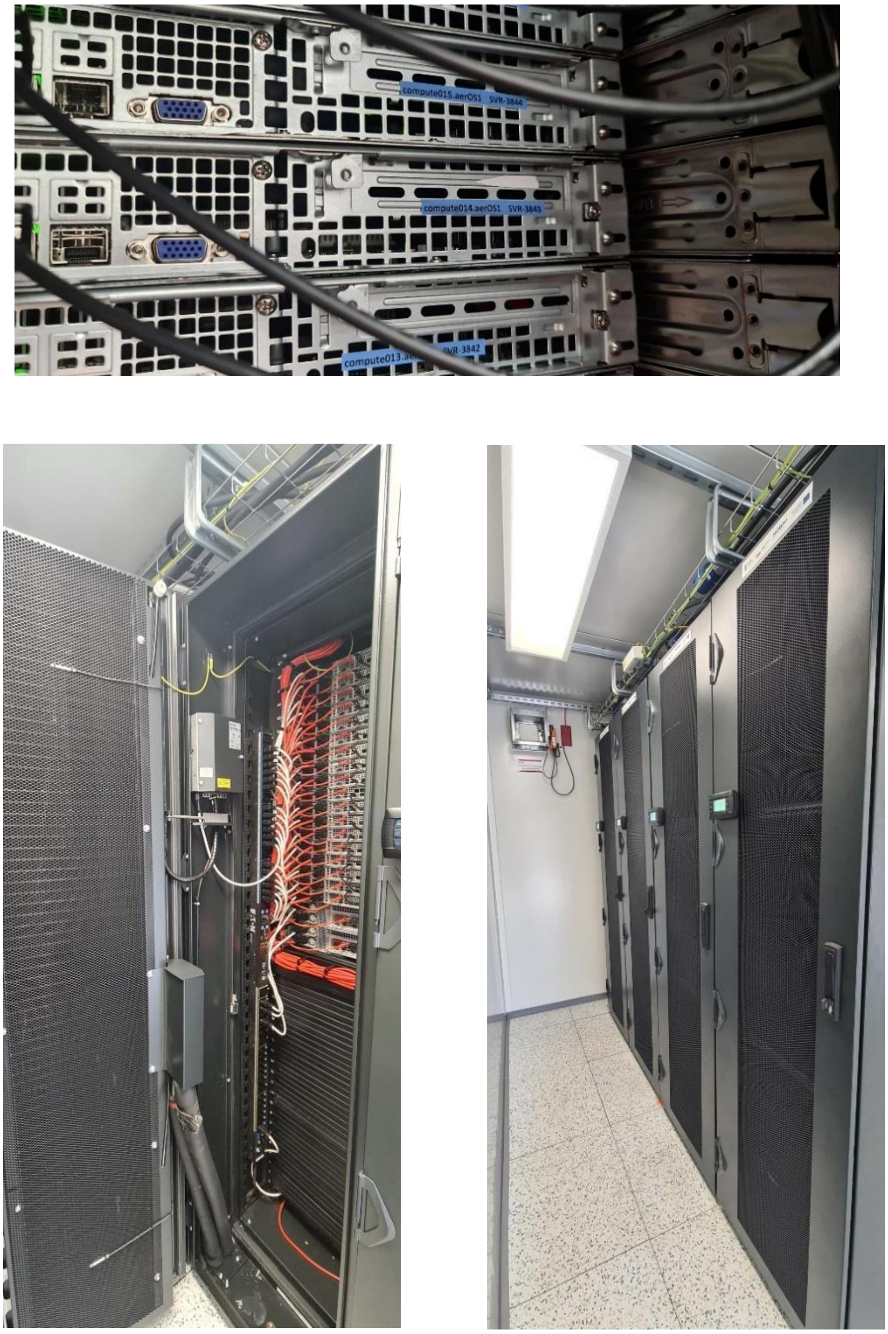


Figure 105. Hardware mounted on the CF container

Figure 106. Cooling system installed near the CF container

Pilot2 - Business Process 1 - Activity - 5 (P2-BP1-DA5): K8s setup and test

Kubernetes clusters were installed on both nodes and central management cluster was installed on Cloud Ferro's public cloud. All these digital infrastructure became part of the aerOS continuum of pilot 2.

The images below showcase the server list in the k8s clusters: aerOS Central, aerOS1, aerOS2.

```
kklimaszewski:~$ kubectl get nodes
NAME                                     STATUS   ROLES    AGE    VERSION
aeros-central-3c56dd2bb3kx-master-0      Ready    master   482d   v1.23.16
aeros-central-3c56dd2bb3kx-master-1      Ready    master   482d   v1.23.16
aeros-central-aeros-cen-2zk4dheorwzx-node-1   Ready    <none>   146d   v1.23.16
aeros-central-aeros-cen-oipkb6oo4jmj-node-0   Ready    <none>   145d   v1.23.16
aeros-central-aeros-cen-oipkb6oo4jmj-node-6   Ready    <none>   37d    v1.23.16
aeros-central-aeros-cen-oipkb6oo4jmj-node-7   Ready    <none>   26d    v1.23.16
```

```
kklimaszewski:~$ kubectl get nodes
NAME                STATUS   ROLES           AGE    VERSION
aeros1-compute001   Ready    <none>          26h    v1.27.12
aeros1-compute002   Ready    <none>          26h    v1.27.12
aeros1-compute003   Ready    <none>          26h    v1.27.12
aeros1-compute004   Ready    <none>          26h    v1.27.12
aeros1-compute005   Ready    <none>          26h    v1.27.12
aeros1-compute006   Ready    <none>          26h    v1.27.12
aeros1-compute007   Ready    <none>          26h    v1.27.12
aeros1-compute008   Ready    <none>          26h    v1.27.12
aeros1-compute009   Ready    <none>          127d   v1.27.12
aeros1-compute010   Ready    <none>          26h    v1.27.12
aeros1-compute011   Ready    <none>          26h    v1.27.12
aeros1-compute012   Ready    <none>          26h    v1.27.12
aeros1-compute013   Ready    <none>          26h    v1.27.12
aeros1-compute014   Ready    <none>          26h    v1.27.12
aeros1-compute015   Ready    <none>          26h    v1.27.12
aeros1-compute016   Ready    <none>          26h    v1.27.12
aeros1-compute017   Ready    <none>          26h    v1.27.12
aeros1-compute018   Ready    <none>          26h    v1.27.12
aeros1-compute019   Ready    <none>          26h    v1.27.12
aeros1-compute020   Ready    <none>          26h    v1.27.12
aeros1-compute021   Ready    <none>          26h    v1.27.12
aeros1-compute022   Ready    <none>          26h    v1.27.12
aeros1-compute023   Ready    <none>          26h    v1.27.12
aeros1-compute024   Ready    <none>          26h    v1.27.12
aeros1-compute025   Ready    <none>          26h    v1.27.12
aeros1-compute026   Ready    <none>          26h    v1.27.12
aeros1-compute027   Ready    <none>          26h    v1.27.12
aeros1-compute028   Ready    <none>          26h    v1.27.12
aeros1-compute029   Ready    <none>          26h    v1.27.12
aeros1-compute030   Ready    <none>          26h    v1.27.12
aeros1-compute031   Ready    <none>          26h    v1.27.12
aeros1-compute032   Ready    <none>          26h    v1.27.12
aeros1-compute033   Ready    <none>          26h    v1.27.12
aeros1-compute034   Ready    <none>          127d   v1.27.12
aeros1-compute035   Ready    <none>          127d   v1.27.12
aeros1-compute036   Ready    <none>          26h    v1.27.12
aeros1-compute037   Ready    <none>          127d   v1.27.12
control001          Ready    control-plane   492d   v1.29.3
```

```
kklimaszewski:~$ kubectl get nodes
NAME                STATUS   ROLES           AGE    VERSION
aeros2-compute001   Ready    <none>          26h    v1.27.12
aeros2-compute002   Ready    <none>          26h    v1.27.12
aeros2-compute003   Ready    <none>          26h    v1.27.12
aeros2-compute004   Ready    <none>          26h    v1.27.12
aeros2-compute005   Ready    <none>          26h    v1.27.12
aeros2-compute006   Ready    <none>          26h    v1.27.12
aeros2-compute007   Ready    <none>          26h    v1.27.12
aeros2-compute008   Ready    <none>          26h    v1.27.12
aeros2-compute009   Ready    <none>          26h    v1.27.12
aeros2-compute010   Ready    <none>          26h    v1.27.12
aeros2-compute011   Ready    <none>          26h    v1.27.12
aeros2-compute012   Ready    <none>          26h    v1.27.12
aeros2-compute013   Ready    <none>          26h    v1.27.12
aeros2-compute014   Ready    <none>          26h    v1.27.12
aeros2-compute015   Ready    <none>          26h    v1.27.12
aeros2-compute016   Ready    <none>          26h    v1.27.12
aeros2-compute017   Ready    <none>          26h    v1.27.12
aeros2-compute018   Ready    <none>          26h    v1.27.12
aeros2-compute019   Ready    <none>          26h    v1.27.12
aeros2-compute020   Ready    <none>          26h    v1.27.12
aeros2-compute021   Ready    <none>          26h    v1.27.12
aeros2-compute022   Ready    <none>          26h    v1.27.12
aeros2-compute023   Ready    <none>          26h    v1.27.12
aeros2-compute024   Ready    <none>          26h    v1.27.12
aeros2-compute025   Ready    <none>          26h    v1.27.12
aeros2-compute026   Ready    <none>          26h    v1.27.12
aeros2-compute027   Ready    <none>          26h    v1.27.12
aeros2-compute028   Ready    <none>          26h    v1.27.12
aeros2-compute029   Ready    <none>          26h    v1.27.12
aeros2-compute030   Ready    <none>          26h    v1.27.12
aeros2-compute031   Ready    <none>          26h    v1.27.12
aeros2-compute032   Ready    <none>          26h    v1.27.12
aeros2-compute033   Ready    <none>          26h    v1.27.12
aeros2-compute034   Ready    <none>          127d   v1.27.12
aeros2-compute035   Ready    <none>          26h    v1.27.12
aeros2-compute036   Ready    <none>          127d   v1.27.12
aeros2-compute037   Ready    <none>          127d   v1.27.12
control001          Ready    control-plane   488d   v1.29.3
```

Figure 107. Server list in the k8s clusters

Pilot2 - Business Process 1 - Activity - 6 (P2-BP1-DA6): HW installation and run test in the LAB

Several tests were successfully conducted in the LAB environment (showcased in the next figures)

Figure 108. CloudFerro's LAB 1

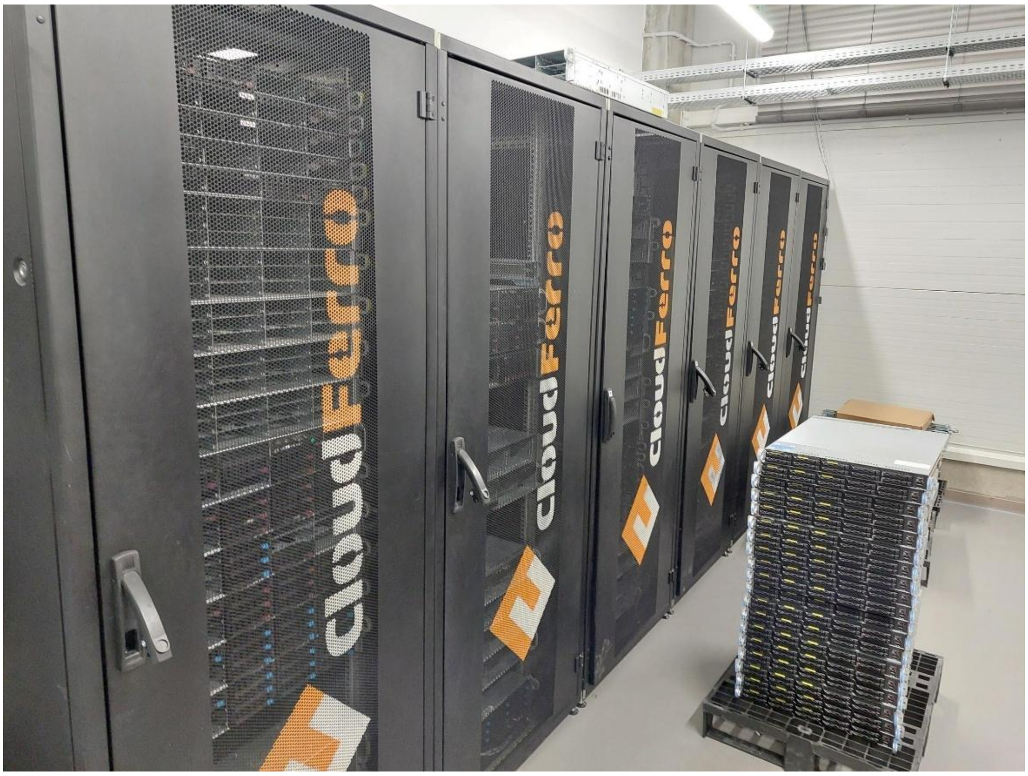


Figure 109. CloudFerro's LAB 2

Pilot2 - Business Process 1 - Activity - 7 (P2-BP1-DA7): Containerized Edge Node Integration with Electrum components

The containerized edge node has been successfully integrated with Electrum system components. Energy meters were connected to the Cloudferro infrastructure, enabling real-time data acquisition and transmission.

Communication between system elements was established using industry-standard protocols: Modbus RTU was applied for data exchange between the PLC and energy meters, while MQTT was used for communication between the PLC and the EMACS platform.

The metering setup enables full bidirectional measurement of both active and reactive energy, capturing energy flows imported from the grid and generated by the photovoltaic installation. Measurements are collected separately for each phase and aggregated for the three-phase system.

To ensure precise energy accounting, internal building consumption (e.g., lighting, HVAC) was separated from the total load. This allows for accurate monitoring of energy used exclusively by the server infrastructure and associated monitoring systems

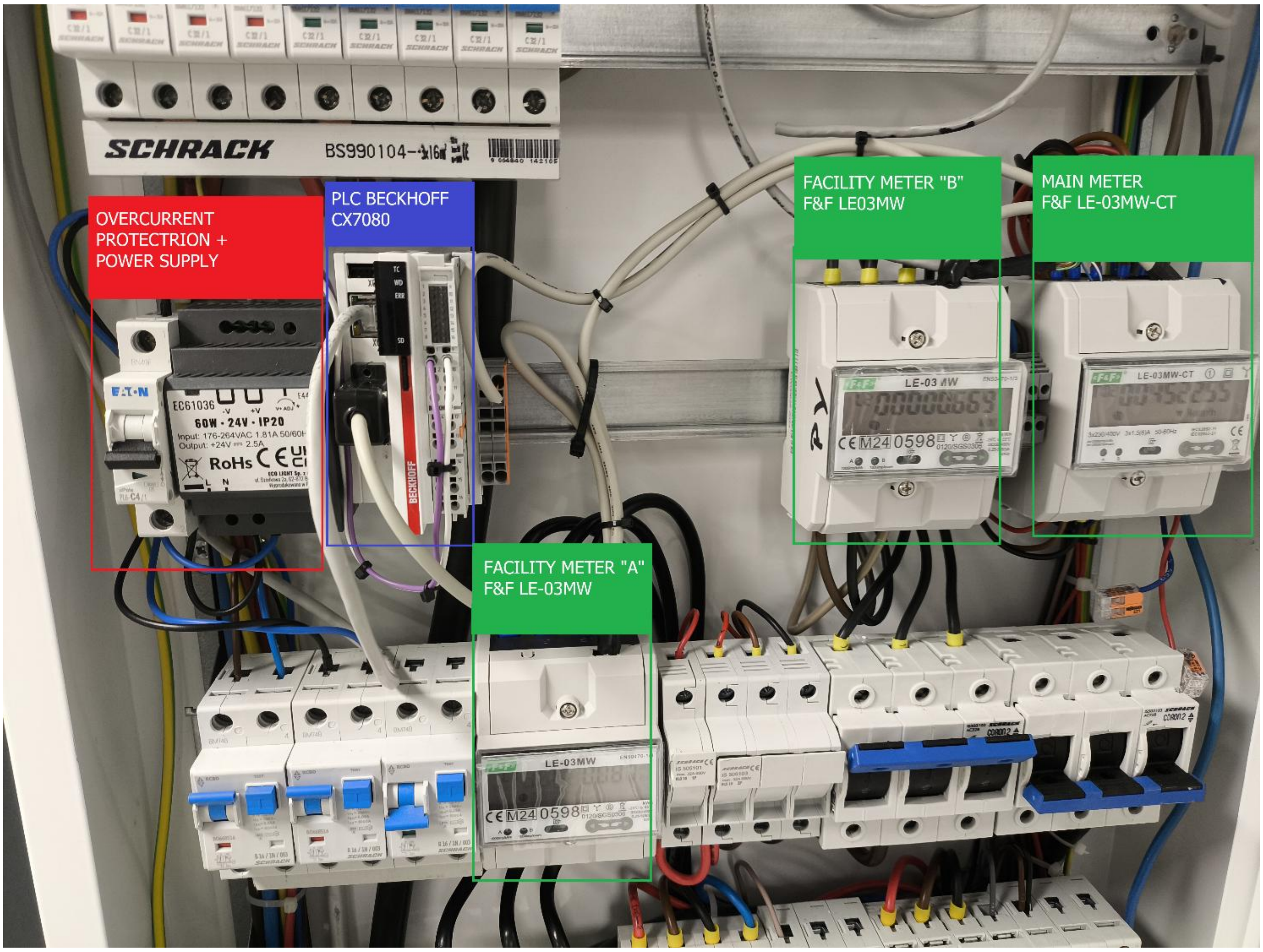


Figure 110. Metering devices for Green Edge Procesing scenario

Objectives and Benefits:

1. Collection and analysis of energy data from energy generation sources to predict the availability of green energy.

2. Collection and analysis of weather data to forecast the operation of the PV farm.

3. Prediction of negative energy prices in short and long-time intervals.

4. Sending control signals regarding the availability period of green Energy at location

Pilot2 - Business Process 1 - Activity - 8 (P2-BP1-DA8): Containerized Edge Node Integration with SRIPAS components

Initial installation of SRIPAS Semantic Annotator and Semantic Translator components was performed during Stage 1: Validation of 1 containerized edge node. At that time, the First Containerized Edge Node was being set up and tested, and core AerOS components were not deployed. Therefore, SRIPAS components were run as standalone components on Kubernetes installed in the initial environment. The installation was done in preparation of a data annotation and translation pipeline. The pipeline was planned to support the data flow from weather station, power sources, and the energy price prediction microservice, into AerOS. The plans were abandoned, in favor of a shorter pipeline, where data flows from the data sources to Orion-LD. The Semantic Annotator and Semantic Translator components were integrated as optional components of the Data Fabric, and the originally planned data pipeline is implemented using only the core components of Data Fabric (see P2-BP1-I11).

Pilot2 - Business Process 1 - Activity - 9 (P2-BP1-DA9): Lab Edge Node Integration with aerOS components

A Kubernetes environment that meets aerOS requirements is set up. Initial versions of basic aerOS components are successfully deployed in this integration environment.

There is network connectivity between clusters. Each cluster has an IP that is accessible from other clusters. For entry point cluster this IP is public. Inside the cluster each node can access others nodes via LAN 10Gb/s.

The following images represent aerOS pods running in lab edge node:

```
kklimaszewski:~$ kubectl get pods -n aeros-management
NAME                                                READY   STATUS      RESTARTS         AGE
aeros-k8s-shim-6dddf7c858-8j5q6                     1/1     Running     837 (2d1h ago)   129d
api-gateway-krakend-56d8f77b45-gl65q                1/1     Running     6 (3d19h ago)    129d
autoscaler-monitor-aeros1-compute-74c46dc554-4bbr5  1/1     Running     0                60m
federator-598cff7445-trw2q                          1/1     Running     911 (47h ago)    129d
hlo-allocator-595577986-fgspq                       1/1     Running     4 (2d1h ago)     16d
hlo-data-aggregator-6b5d5445fc-mb9b9                1/1     Running     4 (3d19h ago)    17d
hlo-deployment-engine-574778d769-tc2nn              1/1     Running     4 (3d19h ago)    15d
hlo-frontend-7c45d4647c-rqbd6                       1/1     Running     4 (2d1h ago)     17d
iota-api-7586fd4f75-9fllw                           1/1     Running     0                15m
iota-dashboard-b656f475d-zrxfp                      1/1     Running     6 (3d19h ago)    129d
iota-hornet-247m6                                   2/2     Running     19 (19h ago)     2d1h
iota-hornet-2rxrc                                   2/2     Running     16 (2d ago)      2d1h
iota-hornet-7ht2g                                   2/2     Running     18 (2d ago)      2d1h
iota-hornet-8jpsk                                   2/2     Running     17 (2d ago)      2d1h
iota-hornet-97wff                                   2/2     Running     17 (2d ago)      2d1h
iota-hornet-d4bpj                                   2/2     Running     17 (2d ago)      2d1h
iota-hornet-dhkvb                                   2/2     Running     17 (2d ago)      2d1h
iota-hornet-dsb4l                                   2/2     Running     17 (2d ago)      2d1h
iota-hornet-f4csf                                   2/2     Running     17 (2d ago)      2d1h
iota-hornet-gqlgm                                   2/2     Running     17 (2d ago)      2d1h
iota-hornet-hb689                                   2/2     Running     17 (2d ago)      2d1h
iota-hornet-hg4rf                                   2/2     Running     12 (2d1h ago)    129d
iota-hornet-jpb6c                                   2/2     Running     16 (2d ago)      2d1h
iota-hornet-lsd2p                                   2/2     Running     18 (2d ago)      2d1h
iota-hornet-n86zk                                   2/2     Running     17 (2d ago)      2d1h
iota-hornet-rs7wt                                   2/2     Running     13913 (2d ago)   128d
iota-hornet-rxhzk                                   2/2     Running     17 (2d ago)      2d1h
iota-hornet-zczsd                                   2/2     Running     17 (2d ago)      2d1h
llo-api-6bdffb476-wvxwq                             1/1     Running     4 (3d19h ago)    17d
orion-ld-broker-7d87d9849c-cjsgd                    1/1     Running     54 (47h ago)     129d
orion-ld-mongodb-0                                  1/1     Running     0                47h
redpanda-0                                          2/2     Running     18 (2d1h ago)    129d
redpanda-configuration-5vskg                        0/1     Completed   0                129d
redpanda-configuration-d5vrk                        0/1     Error       0                129d
redpanda-configuration-pvnlk                        0/1     Error       0                129d
redpanda-console-8668bc994f-g2j8t                   1/1     Running     0                16m
redpanda-post-upgrade-5thm7                         0/1     Error       0                129d
redpanda-post-upgrade-pllms                         0/1     Error       0                129d
redpanda-post-upgrade-q6j9g                         0/1     Completed   0                129d
self-awareness-hardwareinfo-5gwr7                   1/1     Running     39 (31m ago)     18h
self-awareness-hardwareinfo-7hplt                   1/1     Running     40 (8m39s ago)   18h
self-awareness-hardwareinfo-7lbvh                   1/1     Running     47 (8m41s ago)   18h
self-awareness-hardwareinfo-8cmgc                   1/1     Running     39 (8m39s ago)   18h
self-awareness-hardwareinfo-br8qh                   1/1     Running     43 (8m41s ago)   18h
self-awareness-hardwareinfo-d4pgb                   1/1     Running     39 (8m39s ago)   18h
self-awareness-hardwareinfo-gtprj                   1/1     Running     40 (31m ago)     18h
self-awareness-hardwareinfo-gzscz                   1/1     Running     44 (8m39s ago)   18h
self-awareness-hardwareinfo-hf962                   1/1     Running     43 (8m41s ago)   18h
self-awareness-hardwareinfo-k6n26                   1/1     Running     45 (8m39s ago)   18h
self-awareness-hardwareinfo-l5qs6                   1/1     Running     40 (31m ago)     18h
self-awareness-hardwareinfo-mxqrn                   1/1     Running     44 (8m41s ago)   18h
self-awareness-hardwareinfo-pblt4                   1/1     Running     41 (8m42s ago)   18h
self-awareness-hardwareinfo-s9dxs                   1/1     Running     42 (31m ago)     18h
self-awareness-hardwareinfo-w4cg7                   1/1     Running     37 (31m ago)     18h
```

Figure 111. aerOS pods running in lab edge node 1

```
self-awareness-hardwareinfo-ww7rk                      1/1    Running   41 (8m39s ago)   18h
self-awareness-hardwareinfo-xhj2q                      1/1    Running   43 (8m39s ago)   18h
self-awareness-hardwareinfo-z75bf                      1/1    Running   40 (31m ago)     18h
self-awareness-powerconsumptionamd64-26jl6             1/1    Running   15 (31m ago)     18h
self-awareness-powerconsumptionamd64-2czn6             1/1    Running   14 (31m ago)     18h
self-awareness-powerconsumptionamd64-2qdm2             1/1    Running   12 (3h53m ago)   18h
self-awareness-powerconsumptionamd64-44qrw             1/1    Running   14 (3h45m ago)   18h
self-awareness-powerconsumptionamd64-8g58l             1/1    Running   16 (31m ago)     18h
self-awareness-powerconsumptionamd64-f2gr7             1/1    Running   17 (31m ago)     18h
self-awareness-powerconsumptionamd64-h7nv8             1/1    Running   8 (3h45m ago)    18h
self-awareness-powerconsumptionamd64-jf75j             1/1    Running   18 (31m ago)     18h
self-awareness-powerconsumptionamd64-kqnd4             1/1    Running   15 (31m ago)     18h
self-awareness-powerconsumptionamd64-ksdmx             1/1    Running   15 (8m39s ago)   18h
self-awareness-powerconsumptionamd64-l5b6g             1/1    Running   16 (31m ago)     18h
self-awareness-powerconsumptionamd64-lp28q             1/1    Running   17 (62m ago)     18h
self-awareness-powerconsumptionamd64-qqbpb             1/1    Running   13 (31m ago)     18h
self-awareness-powerconsumptionamd64-sbh2w             1/1    Running   10 (31m ago)     18h
self-awareness-powerconsumptionamd64-sp8vn             1/1    Running   13 (31m ago)     18h
self-awareness-powerconsumptionamd64-stxlm             1/1    Running   14 (31m ago)     18h
self-awareness-powerconsumptionamd64-trdvb             1/1    Running   12 (31m ago)     18h
self-awareness-powerconsumptionamd64-whtpn             1/1    Running   17 (8m40s ago)   18h
self-orchestrator-orchestrator-499lg                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-58fhq                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-6tm9w                   1/1    Running   2 (19h ago)      2d1h
self-orchestrator-orchestrator-7jp95                   1/1    Running   5 (2d ago)       128d
self-orchestrator-orchestrator-bh8hx                   1/1    Running   6 (2d1h ago)     129d
self-orchestrator-orchestrator-dhkx6                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-g5vbp                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-hksxj                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-j2lcm                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-kfpgc                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-kzzdz                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-nd5t2                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-q5qgq                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-qwf9g                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-qz82p                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-w7zcx                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-z2jr4                   1/1    Running   1 (2d ago)       2d1h
self-orchestrator-orchestrator-zbk94                   1/1    Running   1 (2d ago)       2d1h
self-security-2bxbx                                    4/4    Running   4 (2d ago)       2d1h
self-security-4kgvm                                    4/4    Running   4 (2d ago)       2d1h
self-security-859lz                                    4/4    Running   20 (2d ago)      128d
self-security-8smrm                                    4/4    Running   4 (2d ago)       2d1h
self-security-8xngb                                    4/4    Running   4 (2d ago)       2d1h
self-security-c6t94                                    4/4    Running   4 (2d ago)       2d1h
self-security-dxknq                                    4/4    Running   4 (2d ago)       2d1h
self-security-ftcr7                                    4/4    Running   4 (2d ago)       2d1h
self-security-ghj7j                                    4/4    Running   4 (2d ago)       2d1h
self-security-gt86q                                    4/4    Running   24 (2d1h ago)    134d
self-security-hj9cm                                    4/4    Running   4 (2d ago)       2d1h
self-security-k5skv                                    4/4    Running   4 (2d ago)       2d1h
self-security-l7dlw                                    4/4    Running   4 (2d ago)       2d1h
self-security-mszz6                                    4/4    Running   4 (2d ago)       2d1h
self-security-pkf8r                                    4/4    Running   4 (2d ago)       2d1h
self-security-pm4cw                                    4/4    Running   4 (2d ago)       2d1h
self-security-rgl9c                                    4/4    Running   8 (19h ago)      2d1h
self-security-tvlrx                                    4/4    Running   4 (2d ago)       2d1h
trustmanager-5b9b8855b4-spplp                          1/1    Running   0                14m
kklimaszewski:~$ kubectl get pods -n aeros-llo-k8s-system
NAME                                                   READY  STATUS    RESTARTS         AGE
llo-k8s-operator-controllermanager-76db797d9-86wws     2/2    Running   46 (47h ago)     7d18h
```

Figure 112. aerOS pods running in lab edge node 2

Pilot2 - Business Process 1 - Activity - 10 (P2-BP1-DA10): Autoscaler monitor Development

Standard aerOS scheduling process for Kubernetes results in creation of Kubernetes Deployment resource strictly assigned to specific Kubernetes node (corresponding to one of IEs). This approach is undesirable when using Kubernetes with cluster-autoscaler since it adds and removes nodes based on load on the cluster. Locking Deployment resources to specific nodes prevents moving workloads between nodes to remove ones with low usage. It also does not work with scaling up since that only happens when kube-scheduler cannot find place for a workload. Such situation does not occur in standard aerOS scheduling.

To remedy that autoscaler-monitor component is developed. It is available at Github[17]. This is a standalone component written in go programming language. It introduces new kind of IE: nodepool IE. Nodepool IE acts as aggregates of standard Kubernetes IEs. When aerOS service component is scheduled to nodepool IE resulting deployment has node selector that matches all nodes in nodepool IE. Available resources reported by nodepool IE are dynamic. They are a sum of resources reported by nodes available in the cluster belonging to the IE but also include potential extra node that could be added by cluster-autoscaler. Each deployment of

[17] https://gitlab.aeros-project.eu/wp3/t3.3/autoscaler-monitor

autoscaler-monitor reports one nodepool IE to orion-ld. Parameters of that nodepool – like resources available on each node, nodepool name, max number of nodes are provided as command line options.

Autoscaler monitor is already operative in our environment. While working it creates and keeps up to date Infrastructure Element like one on the screeshots below (some fields omitted for brevity – those fields are less important for us and are set to their default value (-1, false or empty string):


```
{
  "id": "urn:ngsi-ld:InfrastructureElement:aeros1:aeros1-compute",
  "type": "InfrastructureElement",
  "availableRam": {
    "type": "Property",
    "value": 256308,
    "unitCode": "Megabytes (MB)"
  },
  "avgPowerConsumption": {
    "type": "Property",
    "value": 1495,
    "unitCode": "Watts (W)"
  },
  "containerTechnology": {
    "type": "Property",
    "value": "Kubernetes"
  },
  "cpuArchitecture": {
    "type": "Property",
    "value": "urn:ngsi-ld:CpuArchitecture:x86_64"
  },
  "cpuCores": {
    "type": "Property",
    "value": 86
  },
  "currentCpuUsage": {
    "type": "Property",
    "value": 45,
    "unitCode": "Percentage (%)"
  },
  "currentPowerConsumption": {
    "type": "Property",
    "value": 369,
    "unitCode": "Watts (W)"
  },
  "currentRamUsage": {
    "type": "Property",
    "value": 210941,
    "unitCode": "Megabytes (MB)"
  },
  "currentRamUsagePct": {
    "type": "Property",
    "value": 46,
    "unitCode": "Percentage (%)"
  },
  "diskType": {
    "type": "Property",
    "value": "ssd"
  },
  "domain": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:Domain:aeros1"
  },
```

```
  "energyEfficiencyRatio": {
    "type": "Property",
    "value": 0
  },
  "hostname": {
    "type": "Property",
    "value": "nodepool-aeros1-compute"
  },
  "infrastructureElementStatus": {
    "type": "Property",
    "value": "urn:ngsi-ld:InfrastructureElementStatus:Ready"
  },
  "infrastructureElementTier": {
    "type": "Property",
    "value": "urn:ngsi-ld:InfrastructureElementTier:Cloud"
  },
  "location": {
    "type": "GeoProperty",
    "value": {
      "coordinates": [
        0,
        0
      ],
      "type": "Point"
    }
  },
  "lowLevelOrchestrator": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:LowLevelOrchestrator:aeros1:Kubernetes"
  },
  "operatingSystem": {
    "type": "Property",
    "value": "urn:ngsi-ld:OperatingSystem:linux"
  },
  "powerSource": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:none"
  },
  "ramCapacity": {
    "type": "Property",
    "value": 467249,
    "unitCode": "Megabytes (MB)"
  }
}
```


Figure 113. Autoscaler monitor

Scheduling to such IE is as simple as passing its ID in the TOSCA definition, as shown in the screenshot below:

```yaml
description: test-2025-08-01_001
node_templates:
  sc:
    artifacts:
      application_image:
        file: pause
        repository: k8s.gcr.io
        type: tosca.artifacts.Deployment.Image.Container.Docker
    interfaces:
      Standard:
        create:
          implementation: application_image
          inputs:
            cliArgs: []
            envVars: []
    isJob: false
    requirements:
    - host:
        node_filter:
          properties:
            id:
            - urn:ngsi-ld:InfrastructureElement:aeros1:aeros1-compute
    type: tosca.nodes.Container.Application
serviceOverlay: false
tosca_definitions_version: tosca_simple_yaml_1_3
```


Figure 114. Example of IE ID specified in the TOSCA definition

Produced deployment is scheduled based on "aeros.cloudferro.com/nodepool" selector which all nodes in nodepool specify as label, as seen below (some fields omitted for brevity):


```yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: "2025-08-01T07:23:13Z"
  generateName: aeros-service-6emxkqkq24x0-component-sc-774bc864f-
  labels:
    app.kubernetes.io/created-by: urn_ngsi-ld_LowLevelOrchestrator_aeros1_Kubernetes
    app.kubernetes.io/instance: urn_ngsi-ld_Service_6emxkqkq24x0_Component_sc
    app.kubernetes.io/managed-by: aeros-project.eu
    app.kubernetes.io/name: aeros-service-6emxkqkq24x0-component-sc
    app.kubernetes.io/part-of: urn_ngsi-ld_Service_6emxkqkq24x0
    pod-template-hash: 774bc864f
  name: aeros-service-6emxkqkq24x0-component-sc-774bc864f-dnvd4
  namespace: default
  ownerReferences:
  - apiVersion: apps/v1
    blockOwnerDeletion: true
    controller: true
    kind: ReplicaSet
    name: aeros-service-6emxkqkq24x0-component-sc-774bc864f
    uid: 1baea456-2b97-41de-98c9-370646f1ee5a
  resourceVersion: "267436085"
  uid: 5808bcd0-4591-4aea-8b8f-15f10b08e16a
spec:
  containers:
  - image: k8s.gcr.io/pause
    imagePullPolicy: Always
    name: aeros-service-6emxkqkq24x0-component-sc
  nodeName: aeros1-compute005
  nodeSelector:
    aeros.cloudferro.com/nodepool: aeros1-compute
...
```


Figure 115. Development scheduling based on nodepool label

Pilot2 - Business Process 1 - Activity - 11 (P2-BP1-DA11): Development of future Energy Price microservices

Frugal AI refers to training and inference under tight resource constraints—minimizing data volume, compute, and energy—through model simplicity, focused datasets, and edge deployment strategies. In the current implementation, training is performed exclusively on historical Polish TGE RDN (Next Day Energy Price) data, using a concise set of time-series features. The resulting LSTM model is deployed in a Docker container on an NVIDIA Jetson Orin device at the edge, achieving sub-200 ms predictions under limited compute budgets.

The architecture is modular and containerized, enabling extensibility to other national energy markets. Future edge nodes in various European countries can locally retrain on market-specific data and transmit only neural network weight updates to a central aggregator. This approach enables true Federated Learning—collaborative training of a pan-European model without exchanging raw data, thereby preserving data privacy and reducing network overhead.

System development began with the specification of real-time data collection and forecasting requirements for the Polish TGE RDN market. FastAPI was selected as the framework for both data scraping and prediction services. The scraper service, implemented with FastAPI, periodically retrieves RDN pricing data from TGE.pl and stores raw values in a PostgreSQL v17.3 database, configured with time-series optimizations.

For model development, historical prices were exported from PostgreSQL and enriched with engineered time features to train a TensorFlow LSTM model. After multiple experimental iterations, the model achieved over 85% validation accuracy. The trained model was serialized as a keras artifact.

A dedicated FastAPI-based prediction service was implemented to load the serialized model at startup, retrieve the most recent features from PostgreSQL, and deliver 24-hour forecasts. Both services, along with the PostgreSQL instance, were containerized using Docker v28.1.1. Orchestration is handled through a docker-compose.yml file, which manages volumes, networks, and environment variables, enabling seamless single-command deployment.

Validation mechanisms were applied to the scraping logic. All database interactions are secured using SQLAlchemy ORM. Monitoring is conducted via Grafana dashboards connected to PostgreSQL, which display historical price trends, model predictions, and key performance indicators. Dependency and environment management is handled using Poetry v2.1.3 to ensure versioning and reproducibility.

The prediction API is queried hourly (with configurable frequency). Upon request, the system checks for any changes in relevant input data. If updated data is detected, a new forecast is generated; otherwise, the most recent result is returned. In both cases, data is pushed to Orion LD in the agreed JSON-LD format, ensuring synchronization and uninterrupted data availability.

At present, a continuous evaluation and fine-tuning system is under development. This tool will host the TGE price forecasting use case and include automated drift detection, regular performance evaluation, and periodic retraining. An initial retraining cycle of two weeks is currently being used for testing and development purposes.

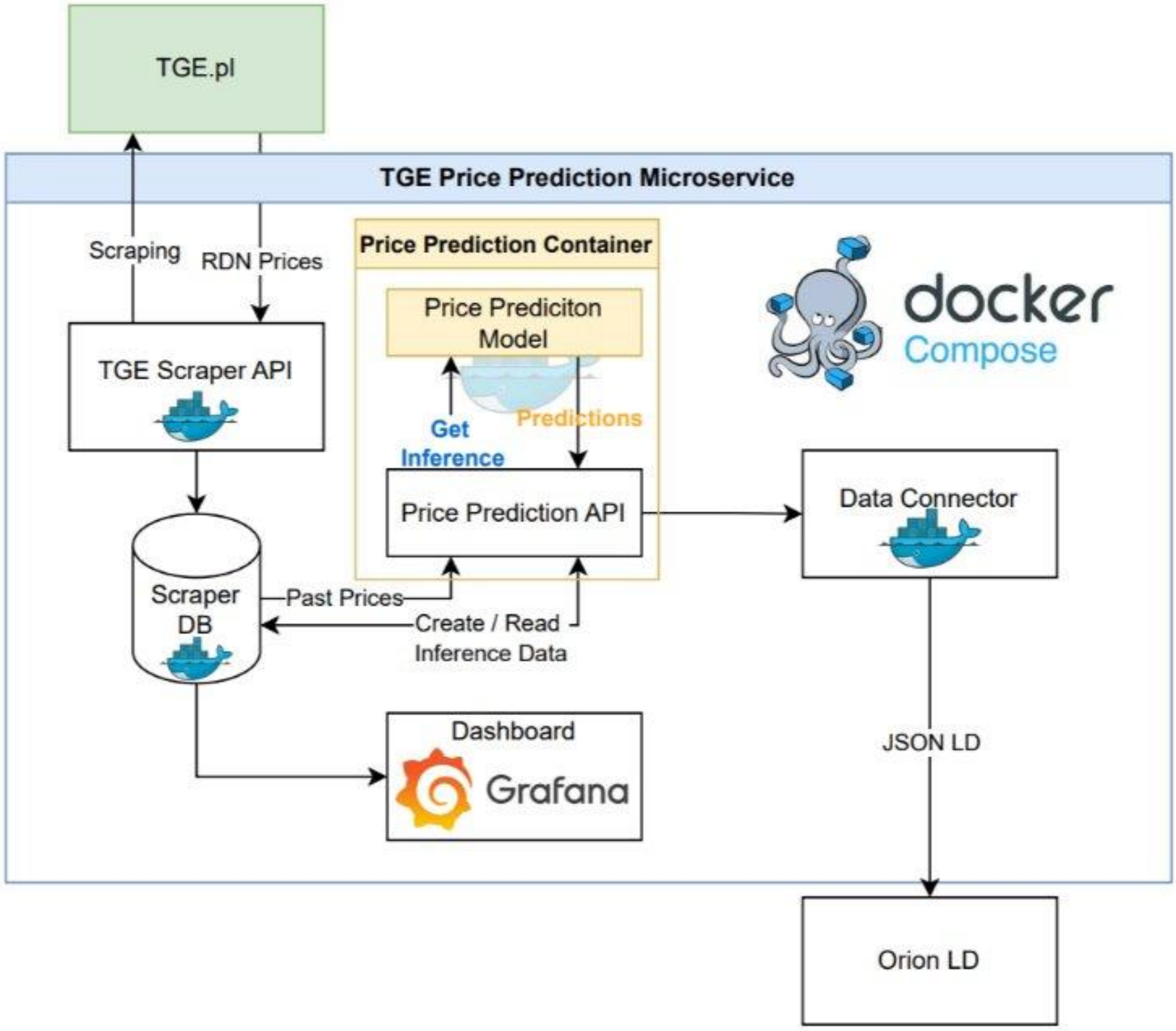


Figure 116. Pilot 2 Future Energy Price Microservice Architecture

Pilot2 - Business Process 1 - Activity - 12 (P2-BP1-DA12): Development of DF the data sources connectors

Dedicated software module has been developed to generate structured energy data packages. The system collects and integrates information from multiple sources, including weather data (current and forecast), energy price predictions, real-time energy consumption and production, and related entities regarding farm and device entities (e.g., weather stations, smart meters).

The data is gathered from both the energy prediction service and the Electrum SCADA system. It is then formatted and transmitted according to the specifications agreed upon with the project partner (SRIPAS), using the NGSI-LD format and Orion LD protocol.

```json
{
    "id": "urn:ngsi-ld:PricePrediction:PL-PricePrediction-ELEC-2025-04-03-123:99",
    "type": "PricePrediction",

    "address": {
        "type": "Property",
        "value": {
            "addressLocality": "ExampleProvince",
            "addressCountry": "PL"
        }
    },
    "location": {
        "type": "GeoProperty",
        "value": {
            "type": "Point",
            "coordinates": [
                21.0122,
                52.2297
            ]
        }
    },

    "dateIssued": "2025-04-01T11:30:00.004033",
    "validFrom": "2025-04-03T02:00:00",
    "validTo": "2025-04-03T03:00:00",

    "https://w3id.org/aerOS/power#f1_price": 393.5117492675781,

    "@context": [
        "https://fiware.github.io/data-models/context.jsonld",
        "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
    ]
}
```

Figure 117. Example Price Prediction NGSI-LD entity used in development of the data sources connectors.

2.2.1.1.3. Integration activities

Pilot2 – Business Process 1 – Activity - 10 (P2-BP1-I10): First Containerized Edge Node Integration with Electrum components

The containerized edge node is connected to Electrum SCADA system.

aerOS basic components are installed as an edge node on the Electrum server and connected to the main Cloud Ferro aerOS instance (acting as entrypoint domain). A service for transferring data from the SCADA system and future energy price is launched.

The following image represents aerOS pods running in Electrum edge node.

```
jetson@k8s-jetson:~$ kubectl get pods -A
NAMESPACE              NAME                                                 READY   STATUS                     RESTARTS        AGE
aeros-llo-api          aeros-llo-api-5cfbbb6f95-rsfw5                       1/1     Running                    289 (45h ago)   119d
aeros-llo-k8s-system   aeros-llo-k8s-controller-manager-78c7bb6f9f-kgjd5    2/2     Running                    314 (45h ago)   116d
default                aeros-k8s-shim-6bd9b59bf4-s2nq2                      1/1     Running                    2 (45h ago)     116d
default                api-gateway-krakend-6c77dbd7b8-n9d4k                 1/1     Running                    1 (45h ago)     18d
default                federator-68ddd648f4-tdwnw                           1/1     Running                    319 (19h ago)   116d
default                hlo-allocator-679fcb74cd-m5k7b                       1/1     Running                    24 (45h ago)    4d4h
default                hlo-allocator-679fcb74cd-pp7zt                       0/1     ContainerStatusUnknown     3               116d
default                hlo-data-aggregator-65c67c59c8-4f2lz                 1/1     Running                    2 (45h ago)     116d
default                hlo-deployment-engine-7985b6ddf9-wdqzm               1/1     Running                    2 (45h ago)     116d
default                hlo-frontend-75c5d7d858-nrcp5                        1/1     Running                    2 (45h ago)     116d
default                iota-api-75455cb847-pj54m                            1/1     Running                    2 (45h ago)     119d
default                iota-dashboard-9f6ff6cfd-b4rq5                       0/1     ContainerStatusUnknown     0               4d4h
default                iota-dashboard-9f6ff6cfd-bgm9d                       0/1     ContainerStatusUnknown     0               4d4h
default                iota-dashboard-9f6ff6cfd-g6pxn                       0/1     ContainerStatusUnknown     39 (4d4h ago)   119d
default                iota-dashboard-9f6ff6cfd-lbzcn                       0/1     ContainerStatusUnknown     0               4d4h
default                iota-dashboard-9f6ff6cfd-mm224                       0/1     ContainerStatusUnknown     0               4d4h
default                iota-dashboard-9f6ff6cfd-rlxhj                       0/1     ContainerStatusUnknown     0               4d4h
default                iota-dashboard-9f6ff6cfd-sfj5p                       1/1     Running                    531 (19h ago)   4d4h
default                iota-dashboard-9f6ff6cfd-xk96r                       0/1     ContainerStatusUnknown     0               4d4h
default                iota-hornet-z6hqj                                    2/2     Running                    315 (19h ago)   119d
default                kpi-grafana-76f6994b75-s692x                         1/1     Running                    33 (45h ago)    97d
default                orion-ld-broker-db979dcdf-xldhj                      1/1     Running                    2 (45h ago)     119d
default                orion-ld-mongodb-0                                   1/1     Running                    2 (45h ago)     119d
default                price-predictor-dataconnector-79c9fc4469-9qlw4       1/1     Running                    2 (45h ago)     103d
default                price-predictor-dataconnector-79c9fc4469-w7djp       1/1     Running                    2 (45h ago)     103d
default                price-predictor-postgres-0                           1/1     Running                    2 (45h ago)     103d
default                price-predictor-priceprediction-6d7f4b96df-dpsq9     1/1     Running                    2 (45h ago)     103d
default                price-predictor-tgescrapper-bd47fcc7-xcr8w           1/1     Running                    2 (45h ago)     103d
default                self-awareness-hardwareinfo-mz9kd                    1/1     Running                    4 (45h ago)     116d
default                self-awareness-powerconsumptionarm64-m6ln6           1/1     Running                    4 (45h ago)     116d
default                self-orchestrator-orchestrator-m4cwp                 1/1     Running                    2 (45h ago)     116d
default                self-security-p5tsm                                  4/4     Running                    8 (45h ago)     116d
ingress-nginx          ingress-nginx-controller-866596f49b-rmkqx            1/1     Running                    331 (45h ago)   18d
kube-flannel           kube-flannel-ds-h7s22                                1/1     Running                    2 (45h ago)     119d
kube-system            coredns-668d6bf9bc-g4j84                             1/1     Running                    47 (45h ago)    119d
kube-system            coredns-668d6bf9bc-qdlbh                             1/1     Running                    46 (45h ago)    119d
kube-system            etcd-k8s-jetson                                      1/1     Running                    14 (45h ago)    119d
kube-system            kube-apiserver-k8s-jetson                            1/1     Running                    125 (45h ago)   119d
kube-system            kube-controller-manager-k8s-jetson                   1/1     Running                    310 (45h ago)   119d
kube-system            kube-proxy-rcklr                                     1/1     Running                    2 (45h ago)     119d
kube-system            kube-scheduler-k8s-jetson                            1/1     Running                    282 (45h ago)   119d
local-path-storage     local-path-provisioner-74f9666bc9-6c44h              1/1     Running                    2 (45h ago)     119d
redpanda               redpanda-0                                           2/2     Running                    179 (45h ago)   119d
redpanda               redpanda-console-6bcdb456b-jq8md                     1/1     Running                    425 (45h ago)   119d
redpanda               redpanda-create-aeros-topics-w9bhn                   0/1     Completed                  0               119d
```

Figure 118. aerOS pods running in Electrum edge node

Pilot2 – Business Process 1 – Activity - 11 (P2-BP1-I11): First Containerized Edge Node Integration with semantic components

Semantic Annotator and Semantic Translator were planned to be used to integrate heterogeneous data sources, and to feed the data into aerOS Meta-OS. After initial integration (see P2-BP1-DA8), the semantic components were successfully integrated into the Data Fabric, as optional components.

Finally, the data connectors for Electrum SCADA system (see P2-BP1-I10) could connect directly to the Data Fabric, without the need for semantic annotation or translation. Similarly, the energy price prediction microservice can pull data directly from relevant sources, and push it to aerOS through the core components of the Data Fabric.

Pilot2 – Business Process 1 – Activity - 12 (P2-BP1-I12): Container deployment

Containers were filled with racks and panels with networking devices and servers. The cooling subsystem was set up. An evaluation of the cooling door technology was successfully conducted.

The proposed cooling system focuses on selective cooling of server components within a containerized environment, rather than cooling the entire container. This approach minimizes energy consumption by targeting heat dissipation directly from the servers. The Rear Door Heat Exchangers (RDHx) facilitate this selective cooling mechanism by strategically extracting heat from the servers and transferring it through the container's rear door. This method ensures efficient thermal management, reducing overall energy requirements and enhancing system reliability.

Figure 119. Pilot 2 Container deployment in Warsaw, CF location

Pilot2 – Business Process 1 – Activity - 13 (P2-BP1-I13): Container connection to PV

Photo Voltaic power added to the main electricity source on CloudFerro test site, as seen in the following image.

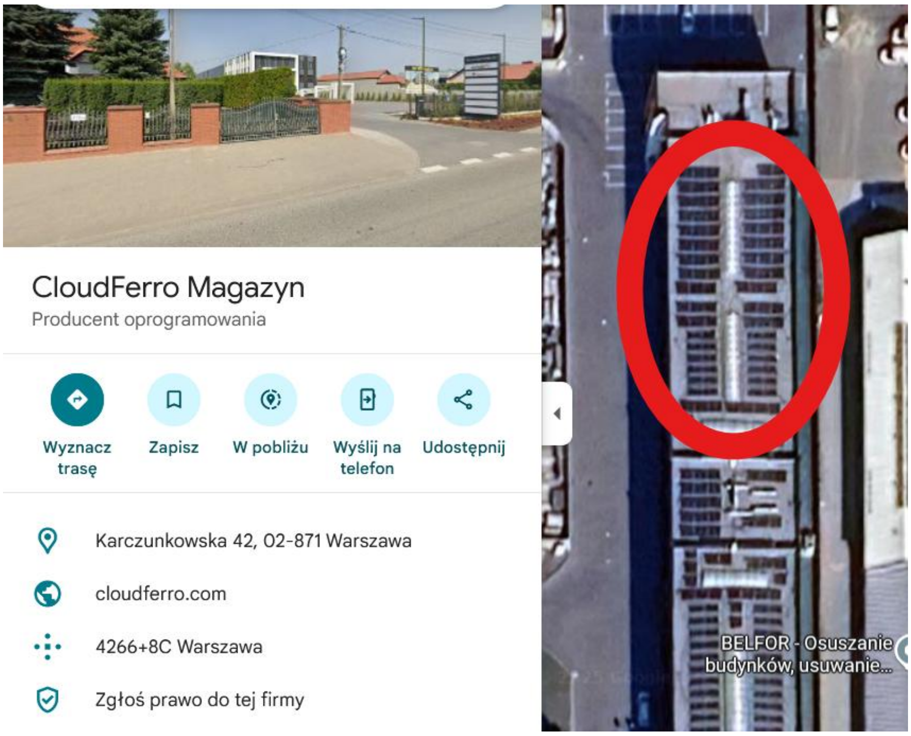


Figure 120. PV infrastructure in CF

Pilot2 – Business Process 2 – Activity - 14 (P2-BP1-I14): Inter-cloud integration

Integration environment was set up in a cloud environment. Cloud and edge (metal containers) aerOS domains are deployed and visible from the entry-point domain.

The connection was established between Electrum Orion (the Context Broker essential for establishing aerOS federation) and the CloudFerro aerOS instance. aerOS was deployed as an edge node, launched it within Kubernetes, and configured it to connect with the CloudFerro aerOS instance following the installation guide available[18].

Pilot2 – Business Process 1 – Activity - 15 (P2-BP1-I15): aerOS Basic components

All aerOS Basic components were installed at current version up-to-date (at the time of this report).

Domains list

Id	Description	Public Url	Owner	Entrypoint	Stc	
aeros-central	This is entrypoint domain running in cloud environment, not for running workloads.	https://entrypoint.pilot2.aeros-project.eu	CloudFerro	✓	Preliminary	
aeros2	This is domain nr 2 for running workloads located in a container.	https://aeros2.aeros.staging.intra.cloudferro.com	CloudFerro	×	Preliminary	
aeros1	This is domain nr 1 for running workloads located in a container.	https://aeros1.aeros.staging.intra.cloudferro.com	CloudFerro	×	Preliminary	
aeros-electrum	Electrum Domain	https://electrum-aeros.com	Electrum	×	Preliminary	View →

Figure 121. Domains list

Domain detail ← Domains

Id	Description	Public Url	Owner	Entrypoint	Status
aeros-central	This is entrypoint domain running in cloud environment, not for running workloads.	https://entrypoint.pilot2.aeros-project.eu	CloudFerro	✓	Preliminary

Infrastructure elements:

Hostname	Container Technology	CPU arch	CPU cores	RAM capacity (MB)	Trust score	Status	Metrics
aeros-central-3c56dd2bb3kx-master-0	Kubernetes	x64	2	7753	-1	Ready	
aeros-central-aeros-cen-2zk4dheorwzx-node-0	Kubernetes	x64	2	7753	-1	Ready	

Figure 122. aerOS central domain details in management portal

Domain detail ← Domains

Id	Description	Public Url	Owner	Entrypoint	Status
aeros-electrum	Electrum Domain	https://electrum-aeros.com	Electrum	×	Preliminary

Infrastructure elements:

Hostname	Container Technology	CPU arch	CPU cores	RAM capacity (MB)	Trust score	Status	Metrics
k8s-jetson	Kubernetes	arm64	8	65894	-1	Ready	

Figure 123. aerOS electrum domain details in management portal

[18] https://docs.aeros-project.eu/en/latest/

Domain detail

← Domains

Id	Description	Public Url	Owner	Entrypoint	Status
aeros1	This is domain nr 1 for running workloads located in a container.	https:// aeros1.aeros.staging.intra.cloudferro.com	CloudFerro	×	Preliminary

Infrastructure elements:

Hostname	Container Technology	CPU arch	CPU cores	RAM capacity (MB)	Trust score	Status	Metrics
aeros1-compute036	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute026	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute016	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute004	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute029	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute001	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute024	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute011	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute013	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute020	Kubernetes	x64	48	202686	-1	Ready	
aeros1-compute023	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute002	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute025	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute015	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute034	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute028	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute019	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute035	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute006	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute027	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute021	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute031	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute022	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute032	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute030	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute009	Kubernetes	x64	48	270332	-1	Ready	
control001	Kubernetes	x64	48	270334	-1	Ready	
aeros1-compute005	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute010	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute018	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute012	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute017	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute037	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute014	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute008	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute007	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute003	Kubernetes	x64	48	270332	-1	Ready	
aeros1-compute033	Kubernetes	x64	48	270332	-1	Ready	

Figure 124. aerOS1 domain details in management portal

Domain detail

Domains

Id	Description	Public Url	Owner	Entrypoint	Status
aeros2	This is domain nr 2 for running workloads located in a container.	https://aeros2.aeros.staging.intra.cloudferro.com	CloudFerro	×	Preliminary

Infrastructure elements:

Hostname	Container Technology	CPU arch	CPU cores	RAM capacity (MB)	Trust score	Status	Metrics
aeros2-compute028	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute022	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute025	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute006	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute032	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute029	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute030	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute023	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute033	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute016	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute020	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute015	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute009	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute027	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute002	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute013	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute021	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute035	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute024	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute012	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute001	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute018	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute031	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute026	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute017	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute003	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute004	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute005	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute019	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute037	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute011	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute008	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute010	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute007	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute034	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute014	Kubernetes	x64	48	270332	-1	Ready	
aeros2-compute036	Kubernetes	x64	48	270332	-1	Ready	
control001	Kubernetes	x64	48	270334	-1	Ready	

Figure 125. aerOS domain details in management portal

Pilot2 – Business Process 1 – Activity - 16 (P2-BP1-I16): aerOS Non-Basic components

To make use of nodepool IEs introduced by the *autoscaler* monitor component, the aerOS component LLO needed to be enhanced to recognize such IEs and make slight changes to applied deployment resources.

For this reason, the team in pilot 2 proceeded to improve aerOS LLO' for Kubernetes, via introuding changes on the official (internal) code repository, concretely on the branch *pilot2/nodepool-ie* of LLO K8s Operator SDK[19] include those patches.

There, Infrastructure Elements with hostname starting with nodepool are given special treatment: Instead of using default label key in nodepool selector, another one is used (specified by command line option) that describes nodepool the node corresponds to.

This results in created Deployment being scheduled only on nodes in selected nodepool. The following group of images validates the installation of aerOS of non basic compoenents on the different domains.

```
kklimaszewski:~$ kubectl get pods -n aeros-management
NAME                                           READY   STATUS    RESTARTS          AGE
aeros-k8s-shim-5f99c95676-mrz2s                1/1     Running   0                 126d
api-gateway-krakend-6c66ff9d7d-tf62j           1/1     Running   0                 129d
federator-585cfb97cf-m7qvk                     1/1     Running   0                 129d
hlo-allocator-98dbb6c79-27pm6                  1/1     Running   0                 16d
hlo-data-aggregator-c77799974-w58bv            1/1     Running   0                 16d
hlo-deployment-engine-5b88cc5dff-bndm6         1/1     Running   0                 15d
hlo-frontend-795dd7cb58-kzw6r                  1/1     Running   0                 16d
iota-api-bc69467f9-bzjgf                       1/1     Running   0                 129d
iota-coordinator-557789f54f-5rjg2              1/1     Running   0                 129d
iota-dashboard-7f6b889f79-p8pf7                1/1     Running   0                 129d
iota-hornet-5fc748ccd6-4j9wr                   1/1     Running   0                 129d
keycloak-idm-database-0                        1/1     Running   0                 126d
keycloak-idm-keycloak-5d79f95457-tqfjs         1/1     Running   0                 132d
llo-api-6d575586c-ntcwx                        1/1     Running   0                 17d
management-portal-backend-744764ff65-d6vr5     1/1     Running   0                 129d
management-portal-frontend-7c94b9f648-xp6b9    1/1     Running   0                 129d
openldap-0                                     1/1     Running   0                 12d
openldap-ltb-passwd-8468d87b4f-jw5km           1/1     Running   0                 177d
openldap-phpldapadmin-5cf757fc9b-5njx2         1/1     Running   0                 177d
orion-ld-broker-54ffcb6d6c-p5688               1/1     Running   5 (14d ago)       129d
orion-ld-mongodb-0                             1/1     Running   0                 126d
redpanda-0                                     2/2     Running   1 (30d ago)       145d
redpanda-console-6d6587b6c8-zrlsl              1/1     Running   234 (146d ago)    177d
self-awareness-hardwareinfo-llhgh              1/1     Running   4 (31d ago)       121d
self-awareness-powerconsumptionamd64-r99zf     1/1     Running   2 (31d ago)       121d
self-orchestrator-orchestrator-2qlqm           1/1     Running   1 (129d ago)      132d
self-security-27ftr                            4/4     Running   0                 26d
self-security-b5gdn                            4/4     Running   0                 132d
self-security-dmss7                            4/4     Running   4 (129d ago)      132d
self-security-g6vcr                            4/4     Running   0                 37d
self-security-h6rx7                            4/4     Running   0                 132d
self-security-tmmnq                            4/4     Running   0                 132d
kklimaszewski:~$ kubectl get pods -n aeros-llo-k8s-system
NAME                                                   READY   STATUS    RESTARTS   AGE
llo-k8s-operator-controllermanager-7c9c9c4988-9q7sq    2/2     Running   0          7d
```

Figure 126. aerOS central domain

[19] https://gitlab.aeros-project.eu/wp3/t3.3/llo-k8s-operator-sdk/-/tree/pilot2/nodepool-ie?ref_type=heads

```
kklimaszewski:~$ kubectl get pods -n aeros-management
NAME                                                READY   STATUS    RESTARTS          AGE
aeros-k8s-shim-6dddf7c858-8j5q6                     1/1     Running   837 (31h ago)     129d
api-gateway-krakend-56d8f77b45-gl65q                1/1     Running   6 (3d1h ago)      129d
autoscaler-monitor-aeros1-compute-74c46dc554-hw6s2  1/1     Running   881 (29h ago)     125d
federator-598cff7445-trw2q                          1/1     Running   911 (29h ago)     129d
hlo-allocator-595577986-fgspq                       1/1     Running   4 (31h ago)       15d
hlo-data-aggregator-6b5d5445fc-mb9b9                1/1     Running   4 (3d1h ago)      16d
hlo-deployment-engine-574778d769-tc2nn              1/1     Running   4 (3d1h ago)      15d
hlo-frontend-7c45d4647c-rqbd6                       1/1     Running   4 (31h ago)       16d
iota-api-7586fd4f75-9hxvd                           1/1     Running   5 (30h ago)       128d
iota-dashboard-b656f475d-zrxfp                      1/1     Running   6 (3d1h ago)      129d
iota-hornet-247m6                                   2/2     Running   19 (115m ago)     31h
iota-hornet-24l2s                                   2/2     Running   18 (30h ago)      31h
iota-hornet-2dllz                                   2/2     Running   24286 (30h ago)   128d
iota-hornet-2rxrc                                   2/2     Running   16 (30h ago)      31h
iota-hornet-46qk9                                   2/2     Running   17 (30h ago)      31h
iota-hornet-4qcjv                                   2/2     Running   18 (30h ago)      31h
iota-hornet-724r2                                   2/2     Running   18 (30h ago)      31h
iota-hornet-75xgh                                   2/2     Running   17 (30h ago)      31h
iota-hornet-7bm6t                                   2/2     Running   18 (30h ago)      31h
iota-hornet-7ht2g                                   2/2     Running   18 (30h ago)      31h
iota-hornet-8jpsk                                   2/2     Running   17 (30h ago)      31h
iota-hornet-97wff                                   2/2     Running   17 (30h ago)      31h
iota-hornet-9r8qt                                   2/2     Running   18 (30h ago)      31h
iota-hornet-b8dzl                                   2/2     Running   23402 (30h ago)   128d
iota-hornet-bp5bh                                   2/2     Running   18 (30h ago)      31h
iota-hornet-d4bpj                                   2/2     Running   17 (30h ago)      31h
iota-hornet-dhkvb                                   2/2     Running   17 (30h ago)      31h
iota-hornet-dsb4l                                   2/2     Running   17 (30h ago)      31h
iota-hornet-f4csf                                   2/2     Running   17 (30h ago)      31h
iota-hornet-fgldh                                   2/2     Running   18 (30h ago)      31h
iota-hornet-fqq97                                   2/2     Running   17 (30h ago)      31h
iota-hornet-gqlgm                                   2/2     Running   17 (30h ago)      31h
iota-hornet-hb689                                   2/2     Running   17 (30h ago)      31h
iota-hornet-hg4rf                                   2/2     Running   12 (31h ago)      129d
iota-hornet-hggc9                                   2/2     Running   18 (30h ago)      31h
iota-hornet-jpb6c                                   2/2     Running   16 (30h ago)      31h
iota-hornet-k86ht                                   2/2     Running   18 (30h ago)      31h
iota-hornet-lsd2p                                   2/2     Running   18 (30h ago)      31h
iota-hornet-mzht5                                   2/2     Running   18 (30h ago)      31h
iota-hornet-n86zk                                   2/2     Running   17 (30h ago)      31h
iota-hornet-nshxj                                   2/2     Running   18 (30h ago)      31h
iota-hornet-qbl5f                                   2/2     Running   17 (30h ago)      31h
iota-hornet-qfzkr                                   2/2     Running   25190 (30h ago)   128d
iota-hornet-rs7wt                                   2/2     Running   13913 (30h ago)   127d
iota-hornet-rxhzk                                   2/2     Running   17 (30h ago)      31h
iota-hornet-tthgz                                   2/2     Running   17 (30h ago)      31h
iota-hornet-w9tlk                                   2/2     Running   18 (30h ago)      31h
iota-hornet-zczsd                                   2/2     Running   17 (30h ago)      31h
```

```
llo-api-6bdffb476-wvxwq                       1/1   Running     4 (3d1h ago)    17d
orion-ld-broker-7d87d9849c-cjsgd              1/1   Running     54 (29h ago)    129d
orion-ld-mongodb-0                            1/1   Running     0               30h
redpanda-0                                    2/2   Running     18 (31h ago)    129d
redpanda-configuration-5vskg                  0/1   Completed   0               129d
redpanda-configuration-d5vrk                  0/1   Error       0               129d
redpanda-configuration-pvnlk                  0/1   Error       0               129d
redpanda-console-8668bc994f-8qn4v             1/1   Running     29 (29h ago)    3d5h
redpanda-post-upgrade-5thm7                   0/1   Error       0               129d
redpanda-post-upgrade-pllms                   0/1   Error       0               129d
redpanda-post-upgrade-q6j9g                   0/1   Completed   0               129d
self-awareness-hardwareinfo-2n8s8             1/1   Running     1 (13m ago)     25m
self-awareness-hardwareinfo-2v84k             1/1   Running     2 (13m ago)     23m
self-awareness-hardwareinfo-5gwr7             1/1   Running     2 (13m ago)     23m
self-awareness-hardwareinfo-6ppwk             1/1   Running     1 (13m ago)     24m
self-awareness-hardwareinfo-7hplt             1/1   Running     2 (13m ago)     25m
self-awareness-hardwareinfo-7lbvh             1/1   Running     2 (13m ago)     25m
self-awareness-hardwareinfo-8cmgc             1/1   Running     2 (13m ago)     23m
self-awareness-hardwareinfo-br8qh             1/1   Running     2 (13m ago)     23m
self-awareness-hardwareinfo-cc4gv             1/1   Running     2 (13m ago)     25m
self-awareness-hardwareinfo-d4pgb             1/1   Running     1 (13m ago)     23m
self-awareness-hardwareinfo-dkcv2             1/1   Running     1 (13m ago)     25m
self-awareness-hardwareinfo-dm9kl             1/1   Running     1 (13m ago)     24m
self-awareness-hardwareinfo-dwxfn             1/1   Running     2 (13m ago)     24m
self-awareness-hardwareinfo-gtprj             1/1   Running     1 (13m ago)     23m
self-awareness-hardwareinfo-gzscz             1/1   Running     1 (13m ago)     23m
self-awareness-hardwareinfo-hf962             1/1   Running     2 (13m ago)     24m
self-awareness-hardwareinfo-hk7mv             1/1   Running     1 (13m ago)     23m
self-awareness-hardwareinfo-jv9m6             1/1   Running     2 (13m ago)     25m
self-awareness-hardwareinfo-k6n26             1/1   Running     2 (13m ago)     24m
self-awareness-hardwareinfo-ksbl2             1/1   Running     2 (13m ago)     24m
self-awareness-hardwareinfo-kttjl             1/1   Running     1 (13m ago)     25m
self-awareness-hardwareinfo-l5qs6             1/1   Running     2 (13m ago)     23m
self-awareness-hardwareinfo-lbfpw             1/1   Running     1 (13m ago)     24m
self-awareness-hardwareinfo-mnrdb             1/1   Running     0               24m
self-awareness-hardwareinfo-mxqrn             1/1   Running     2 (13m ago)     25m
self-awareness-hardwareinfo-pblt4             1/1   Running     2 (13m ago)     24m
self-awareness-hardwareinfo-pksbp             1/1   Running     2 (13m ago)     24m
self-awareness-hardwareinfo-rqhb8             1/1   Running     1 (13m ago)     24m
self-awareness-hardwareinfo-s9dxs             1/1   Running     2 (13m ago)     23m
self-awareness-hardwareinfo-tjsfn             1/1   Running     1 (13m ago)     25m
self-awareness-hardwareinfo-vzhx2             1/1   Running     1 (13m ago)     24m
self-awareness-hardwareinfo-w4cg7             1/1   Running     1 (13m ago)     25m
self-awareness-hardwareinfo-ww7rk             1/1   Running     1 (13m ago)     23m
self-awareness-hardwareinfo-x2bsm             1/1   Running     2 (13m ago)     24m
self-awareness-hardwareinfo-x8v4c             1/1   Running     1 (13m ago)     23m
self-awareness-hardwareinfo-xhj2q             1/1   Running     2 (13m ago)     23m
self-awareness-hardwareinfo-z75bf             1/1   Running     0               25m
self-awareness-hardwareinfo-zkb4t             1/1   Running     1 (13m ago)     24m
self-awareness-powerconsumptionamd64-26jl6    1/1   Running     2 (13m ago)     19m
self-awareness-powerconsumptionamd64-2czn6    1/1   Running     1 (13m ago)     23m
self-awareness-powerconsumptionamd64-2qdm2    1/1   Running     0               21m
```

```
self-awareness-powerconsumptionamd64-44qrw        1/1     Running    0                 18m
self-awareness-powerconsumptionamd64-4skpc        1/1     Running    0                 22m
self-awareness-powerconsumptionamd64-6tlzw        1/1     Running    0                 24m
self-awareness-powerconsumptionamd64-82r5q        1/1     Running    0                 25m
self-awareness-powerconsumptionamd64-8g58l        1/1     Running    0                 21m
self-awareness-powerconsumptionamd64-9mx8g        1/1     Running    0                 23m
self-awareness-powerconsumptionamd64-f2gr7        1/1     Running    0                 25m
self-awareness-powerconsumptionamd64-fntdd        1/1     Running    0                 20m
self-awareness-powerconsumptionamd64-frgws        1/1     Running    0                 19m
self-awareness-powerconsumptionamd64-g8m4t        1/1     Running    0                 24m
self-awareness-powerconsumptionamd64-h7nv8        1/1     Running    0                 20m
self-awareness-powerconsumptionamd64-hbqhs        1/1     Running    0                 25m
self-awareness-powerconsumptionamd64-hf4xb        1/1     Running    0                 19m
self-awareness-powerconsumptionamd64-jf75j        1/1     Running    0                 21m
self-awareness-powerconsumptionamd64-jk7cg        1/1     Running    0                 17m
self-awareness-powerconsumptionamd64-jxz8l        1/1     Running    2 (13m ago)       18m
self-awareness-powerconsumptionamd64-kqnd4        1/1     Running    0                 24m
self-awareness-powerconsumptionamd64-ksdmx        1/1     Running    2 (13m ago)       18m
self-awareness-powerconsumptionamd64-l5b6g        1/1     Running    0                 17m
self-awareness-powerconsumptionamd64-lp28q        1/1     Running    0                 17m
self-awareness-powerconsumptionamd64-mrkgh        1/1     Running    0                 25m
self-awareness-powerconsumptionamd64-pbdvz        1/1     Running    0                 19m
self-awareness-powerconsumptionamd64-pp7wh        1/1     Running    0                 18m
self-awareness-powerconsumptionamd64-qqbpb        1/1     Running    0                 18m
self-awareness-powerconsumptionamd64-sbh2w        1/1     Running    0                 22m
self-awareness-powerconsumptionamd64-snvjd        1/1     Running    0                 17m
self-awareness-powerconsumptionamd64-sp8vn        1/1     Running    0                 25m
self-awareness-powerconsumptionamd64-stxlm        1/1     Running    0                 20m
self-awareness-powerconsumptionamd64-t5jfs        1/1     Running    0                 23m
self-awareness-powerconsumptionamd64-trdvb        1/1     Running    0                 17m
self-awareness-powerconsumptionamd64-twxzn        1/1     Running    0                 21m
self-awareness-powerconsumptionamd64-vnp7f        1/1     Running    1 (13m ago)       20m
self-awareness-powerconsumptionamd64-vp5hs        1/1     Running    0                 24m
self-awareness-powerconsumptionamd64-whtpn        1/1     Running    2 (13m ago)       22m
self-awareness-powerconsumptionamd64-zhng7        1/1     Running    0                 25m
self-orchestrator-orchestrator-499lg              1/1     Running    1 (30h ago)       31h
self-orchestrator-orchestrator-58fhq              1/1     Running    1 (30h ago)       31h
self-orchestrator-orchestrator-6k7jn              1/1     Running    1 (30h ago)       31h
self-orchestrator-orchestrator-6tm9w              1/1     Running    2 (115m ago)      31h
self-orchestrator-orchestrator-77rvt              1/1     Running    1 (30h ago)       31h
self-orchestrator-orchestrator-7jp95              1/1     Running    5 (30h ago)       127d
self-orchestrator-orchestrator-7lpc5              1/1     Running    1 (30h ago)       31h
self-orchestrator-orchestrator-bh8hx              1/1     Running    6 (31h ago)       129d
self-orchestrator-orchestrator-d9vpv              1/1     Running    5 (30h ago)       128d
self-orchestrator-orchestrator-dhkx6              1/1     Running    1 (30h ago)       31h
self-orchestrator-orchestrator-g5vbp              1/1     Running    1 (30h ago)       31h
self-orchestrator-orchestrator-hksxj              1/1     Running    1 (30h ago)       31h
self-orchestrator-orchestrator-j2lcm              1/1     Running    1 (30h ago)       31h
self-orchestrator-orchestrator-jbr2h              1/1     Running    1 (30h ago)       31h
self-orchestrator-orchestrator-jwb25              1/1     Running    1 (30h ago)       31h
self-orchestrator-orchestrator-kfp7j              1/1     Running    1 (30h ago)       31h
self-orchestrator-orchestrator-kfpgc              1/1     Running    1 (30h ago)       31h
```

```
self-orchestrator-orchestrator-kzzdz          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-lxs7z          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-nd5t2          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-nqfdp          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-ns2vm          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-q5kqt          1/1     Running     5 (30h ago)      128d
self-orchestrator-orchestrator-q5qgq          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-q9s2t          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-qwf9g          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-qz82p          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-r2kj8          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-rwktb          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-rz6jt          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-slxm7          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-smdfd          1/1     Running     5 (30h ago)      128d
self-orchestrator-orchestrator-w7zcx          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-wxhc7          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-xl4cj          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-z2jr4          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-zbk94          1/1     Running     1 (30h ago)      31h
self-orchestrator-orchestrator-zkdl4          1/1     Running     1 (30h ago)      31h
self-security-28z2l                           4/4     Running     4 (30h ago)      31h
self-security-2bxbx                           4/4     Running     4 (30h ago)      31h
self-security-4jdxh                           4/4     Running     20 (30h ago)     128d
self-security-4kgvm                           4/4     Running     4 (30h ago)      31h
self-security-5x2wq                           4/4     Running     4 (30h ago)      31h
self-security-6hxgp                           4/4     Running     4 (30h ago)      31h
self-security-6qq6s                           4/4     Running     4 (30h ago)      31h
self-security-859lz                           4/4     Running     20 (30h ago)     127d
self-security-8smrm                           4/4     Running     4 (30h ago)      31h
self-security-8xngb                           4/4     Running     4 (30h ago)      31h
self-security-bmb5p                           4/4     Running     4 (30h ago)      31h
self-security-c6t94                           4/4     Running     4 (30h ago)      31h
self-security-dxknq                           4/4     Running     4 (30h ago)      31h
self-security-ftcr7                           4/4     Running     4 (30h ago)      31h
self-security-ghj7j                           4/4     Running     4 (30h ago)      31h
self-security-gt86q                           4/4     Running     24 (31h ago)     133d
self-security-hj9cm                           4/4     Running     4 (30h ago)      31h
self-security-hl7vx                           4/4     Running     4 (30h ago)      31h
self-security-jnzs9                           4/4     Running     4 (30h ago)      31h
self-security-k5skv                           4/4     Running     4 (30h ago)      31h
self-security-kcqxm                           4/4     Running     4 (30h ago)      31h
self-security-kq2mg                           4/4     Running     4 (30h ago)      31h
self-security-l7dlw                           4/4     Running     4 (30h ago)      31h
self-security-mszz6                           4/4     Running     4 (30h ago)      31h
self-security-mzlnb                           4/4     Running     4 (30h ago)      31h
self-security-nkcfh                           4/4     Running     4 (30h ago)      31h
self-security-nq8b4                           4/4     Running     4 (30h ago)      31h
self-security-pkf8r                           4/4     Running     4 (30h ago)      31h
self-security-pm4cw                           4/4     Running     4 (30h ago)      31h
self-security-qj8k8                           4/4     Running     4 (30h ago)      31h
self-security-rgl9c                           4/4     Running     8 (115m ago)     31h
self-security-tg7kk                           4/4     Running     4 (30h ago)      31h
self-security-tvlrx                           4/4     Running     4 (30h ago)      31h
self-security-v8pn5                           4/4     Running     20 (30h ago)     128d
self-security-vd7ms                           4/4     Running     4 (30h ago)      31h
self-security-vfjvl                           4/4     Running     4 (30h ago)      31h
self-security-w69zp                           4/4     Running     4 (30h ago)      31h
self-security-zc7d7                           4/4     Running     20 (30h ago)     128d
trustmanager-5b9b8855b4-lmvz8                 1/1     Running     0                108m
kklimaszewski:~$ kubectl get pods -n aeros-llo-k8s-system
NAME                                                 READY   STATUS    RESTARTS       AGE
llo-k8s-operator-controllermanager-76db797d9-86wws   2/2     Running   46 (29h ago)   7d
```

Figure 127. aerOS 1 Domain

```
kklimaszewski:~$ kubectl get pods -n aeros-management
NAME                                                READY   STATUS    RESTARTS          AGE
aeros-k8s-shim-76dd756c98-z7b8h                     1/1     Running   19067 (31h ago)   129d
api-gateway-krakend-56d8f77b45-n7wrx                1/1     Running   7 (31h ago)       129d
autoscaler-monitor-aeros2-compute-5f98fb6548-frz4t  1/1     Running   19104 (29h ago)   129d
federator-64f78bd6f4-5f5ws                          1/1     Running   19723 (29h ago)   129d
hlo-allocator-595577986-xq54p                       1/1     Running   3 (31h ago)       15d
hlo-data-aggregator-6b5d5445fc-r25kd                1/1     Running   3 (31h ago)       16d
hlo-deployment-engine-574778d769-6bdw8              1/1     Running   3 (31h ago)       15d
hlo-frontend-7c45d4647c-tz5m9                       1/1     Running   3 (31h ago)       16d
iota-api-7586fd4f75-9mht2                           1/1     Running   2 (30h ago)       3d5h
iota-dashboard-b656f475d-jg9g5                      1/1     Running   17693 (31h ago)   129d
iota-hornet-27vp2                                   2/2     Running   19062 (31h ago)   129d
iota-hornet-2gc9z                                   2/2     Running   18 (30h ago)      31h
iota-hornet-2n8dq                                   2/2     Running   18 (30h ago)      31h
iota-hornet-2nzz6                                   2/2     Running   19 (30h ago)      31h
iota-hornet-4vctr                                   2/2     Running   18 (30h ago)      31h
iota-hornet-5g4cv                                   2/2     Running   19 (30h ago)      31h
iota-hornet-5vstk                                   2/2     Running   274 (30h ago)     127d
iota-hornet-6bg2k                                   2/2     Running   18 (30h ago)      31h
iota-hornet-76k5q                                   2/2     Running   19 (30h ago)      31h
iota-hornet-7fcr4                                   2/2     Running   19 (30h ago)      31h
iota-hornet-7wvml                                   2/2     Running   260 (30h ago)     127d
iota-hornet-9n98n                                   2/2     Running   19 (30h ago)      31h
iota-hornet-9rzdh                                   2/2     Running   18 (30h ago)      31h
iota-hornet-bvx9d                                   2/2     Running   19 (30h ago)      31h
iota-hornet-c2k5d                                   2/2     Running   19 (30h ago)      31h
iota-hornet-cbvv4                                   2/2     Running   19 (30h ago)      31h
iota-hornet-cptvx                                   2/2     Running   18 (30h ago)      31h
iota-hornet-d2db5                                   2/2     Running   19 (30h ago)      31h
iota-hornet-fjzrn                                   2/2     Running   18 (30h ago)      31h
iota-hornet-gs4s4                                   2/2     Running   253 (30h ago)     127d
iota-hornet-hw5tf                                   2/2     Running   18 (30h ago)      31h
iota-hornet-j9l8q                                   2/2     Running   19 (30h ago)      31h
iota-hornet-jnr4x                                   2/2     Running   18 (30h ago)      31h
iota-hornet-kdxqz                                   2/2     Running   18 (30h ago)      31h
iota-hornet-kkflc                                   2/2     Running   19 (30h ago)      31h
iota-hornet-l48gg                                   2/2     Running   19 (30h ago)      31h
iota-hornet-ldnjv                                   2/2     Running   19 (30h ago)      31h
iota-hornet-mf94b                                   2/2     Running   19 (30h ago)      31h
iota-hornet-p2sl5                                   2/2     Running   19 (30h ago)      31h
iota-hornet-ph4wb                                   2/2     Running   18 (30h ago)      31h
iota-hornet-ppp52                                   2/2     Running   19 (30h ago)      31h
iota-hornet-sr42j                                   2/2     Running   19 (30h ago)      31h
iota-hornet-st7lr                                   2/2     Running   19 (30h ago)      31h
iota-hornet-tqw24                                   2/2     Running   19 (30h ago)      31h
iota-hornet-vdj48                                   2/2     Running   19 (30h ago)      31h
iota-hornet-vwdv5                                   2/2     Running   19 (30h ago)      31h
iota-hornet-wxj8f                                   2/2     Running   19 (30h ago)      31h
iota-hornet-x57l2                                   2/2     Running   18 (30h ago)      31h
llo-api-6bdffb476-g8gsn                             1/1     Running   3 (31h ago)       17d
orion-ld-broker-cbfb76bcc-ts6wb                     1/1     Running   0                 29h
```

```
orion-ld-mongodb-0                                 1/1   Running     7 (31h ago)     129d
redpanda-0                                         2/2   Running     11 (31h ago)    129d
redpanda-configuration-2cg2w                       0/1   Completed   0               129d
redpanda-configuration-p6cfp                       0/1   Error       0               129d
redpanda-configuration-ssn8d                       0/1   Error       0               129d
redpanda-console-8668bc994f-hfn5f                  1/1   Running     29 (29h ago)    3d5h
redpanda-post-upgrade-4mkbw                        0/1   Error       0               129d
redpanda-post-upgrade-kf5jn                        0/1   Error       0               129d
redpanda-post-upgrade-xdwtw                        0/1   Completed   0               129d
self-awareness-hardwareinfo-22v5v                  1/1   Running     2 (5m23s ago)   28m
self-awareness-hardwareinfo-2dcm2                  1/1   Running     2 (5m26s ago)   28m
self-awareness-hardwareinfo-557n4                  1/1   Running     2 (5m26s ago)   28m
self-awareness-hardwareinfo-6gzjp                  1/1   Running     2 (5m27s ago)   28m
self-awareness-hardwareinfo-7684l                  1/1   Running     2 (5m26s ago)   28m
self-awareness-hardwareinfo-8s5rp                  1/1   Running     2 (5m21s ago)   28m
self-awareness-hardwareinfo-9bmfp                  1/1   Running     2 (5m24s ago)   27m
self-awareness-hardwareinfo-9glqj                  1/1   Running     2 (5m24s ago)   28m
self-awareness-hardwareinfo-b4mv7                  1/1   Running     2 (5m21s ago)   28m
self-awareness-hardwareinfo-cbpbq                  1/1   Running     2 (5m23s ago)   28m
self-awareness-hardwareinfo-cfwz9                  1/1   Running     2 (5m23s ago)   28m
self-awareness-hardwareinfo-cm96q                  1/1   Running     2 (5m21s ago)   28m
self-awareness-hardwareinfo-g89q5                  1/1   Running     2 (5m25s ago)   27m
self-awareness-hardwareinfo-ggtl9                  1/1   Running     2 (5m23s ago)   28m
self-awareness-hardwareinfo-gx7qx                  1/1   Running     2 (5m21s ago)   28m
self-awareness-hardwareinfo-h6zxp                  1/1   Running     2 (5m22s ago)   28m
self-awareness-hardwareinfo-h9snk                  1/1   Running     2 (5m24s ago)   27m
self-awareness-hardwareinfo-hxlkt                  1/1   Running     2 (5m26s ago)   28m
self-awareness-hardwareinfo-jhjmq                  1/1   Running     2 (5m24s ago)   28m
self-awareness-hardwareinfo-kfszv                  1/1   Running     2 (5m22s ago)   27m
self-awareness-hardwareinfo-m9dll                  1/1   Running     2 (5m22s ago)   28m
self-awareness-hardwareinfo-mc2mj                  1/1   Running     2 (5m26s ago)   28m
self-awareness-hardwareinfo-mjbbr                  1/1   Running     2 (5m21s ago)   27m
self-awareness-hardwareinfo-mpwcx                  1/1   Running     2 (5m25s ago)   28m
self-awareness-hardwareinfo-mzwj7                  1/1   Running     2 (5m22s ago)   28m
self-awareness-hardwareinfo-n4rj7                  1/1   Running     2 (5m22s ago)   27m
self-awareness-hardwareinfo-rd6cb                  1/1   Running     2 (5m27s ago)   27m
self-awareness-hardwareinfo-s64dv                  1/1   Running     2 (5m20s ago)   28m
self-awareness-hardwareinfo-sp764                  1/1   Running     2 (5m22s ago)   28m
self-awareness-hardwareinfo-t57fq                  1/1   Running     2 (5m26s ago)   28m
self-awareness-hardwareinfo-vkrgn                  1/1   Running     2 (5m26s ago)   28m
self-awareness-hardwareinfo-vslx9                  1/1   Running     2 (5m24s ago)   28m
self-awareness-hardwareinfo-w28cr                  1/1   Running     2 (5m21s ago)   28m
self-awareness-hardwareinfo-xt7hl                  1/1   Running     2 (5m26s ago)   28m
self-awareness-hardwareinfo-xz9rr                  1/1   Running     2 (5m24s ago)   28m
self-awareness-hardwareinfo-zlvc6                  1/1   Running     2 (5m24s ago)   28m
self-awareness-hardwareinfo-zs2c8                  1/1   Running     2 (5m25s ago)   27m
self-awareness-hardwareinfo-zzpng                  1/1   Running     2 (5m24s ago)   28m
self-awareness-powerconsumptionamd64-28czn         1/1   Running     2 (5m23s ago)   28m
self-awareness-powerconsumptionamd64-49m2j         1/1   Running     2 (5m23s ago)   28m
self-awareness-powerconsumptionamd64-4xp8x         1/1   Running     2 (5m15s ago)   26m
self-awareness-powerconsumptionamd64-5rlzn         1/1   Running     2 (5m18s ago)   28m
self-awareness-powerconsumptionamd64-6gdtr         1/1   Running     2 (5m26s ago)   28m
```

```
self-awareness-powerconsumptionamd64-7sc25          1/1     Running     2 (5m27s ago)     28m
self-awareness-powerconsumptionamd64-97262          1/1     Running     2 (5m26s ago)     28m
self-awareness-powerconsumptionamd64-9nrdm          1/1     Running     2 (5m23s ago)     28m
self-awareness-powerconsumptionamd64-9snm4          1/1     Running     2 (5m21s ago)     28m
self-awareness-powerconsumptionamd64-bgl2m          1/1     Running     1 (5m16s ago)     27m
self-awareness-powerconsumptionamd64-bmc8b          1/1     Running     2 (5m19s ago)     28m
self-awareness-powerconsumptionamd64-dplzv          1/1     Running     0                 28m
self-awareness-powerconsumptionamd64-dqk6q          1/1     Running     2 (5m26s ago)     28m
self-awareness-powerconsumptionamd64-dt6q8          1/1     Running     2 (5m20s ago)     28m
self-awareness-powerconsumptionamd64-h4ddp          1/1     Running     0                 28m
self-awareness-powerconsumptionamd64-hbvkm          1/1     Running     2 (5m18s ago)     25m
self-awareness-powerconsumptionamd64-jg7gx          1/1     Running     2 (5m25s ago)     28m
self-awareness-powerconsumptionamd64-k9b49          1/1     Running     2 (5m17s ago)     27m
self-awareness-powerconsumptionamd64-l457j          1/1     Running     2 (5m17s ago)     27m
self-awareness-powerconsumptionamd64-lcqvg          1/1     Running     0                 25m
self-awareness-powerconsumptionamd64-ljlm6          1/1     Running     2 (5m21s ago)     28m
self-awareness-powerconsumptionamd64-ltgwv          1/1     Running     2 (5m26s ago)     27m
self-awareness-powerconsumptionamd64-mxlt6          1/1     Running     0                 28m
self-awareness-powerconsumptionamd64-p8mt8          1/1     Running     2 (5m18s ago)     25m
self-awareness-powerconsumptionamd64-q2jhg          1/1     Running     0                 28m
self-awareness-powerconsumptionamd64-rdfmr          1/1     Running     2 (5m19s ago)     28m
self-awareness-powerconsumptionamd64-rp9hn          1/1     Running     0                 28m
self-awareness-powerconsumptionamd64-s5nvc          1/1     Running     0                 28m
self-awareness-powerconsumptionamd64-smv92          1/1     Running     0                 28m
self-awareness-powerconsumptionamd64-tcxtx          1/1     Running     2 (5m20s ago)     28m
self-awareness-powerconsumptionamd64-tv6sj          1/1     Running     1 (5m15s ago)     26m
self-awareness-powerconsumptionamd64-v8c87          1/1     Running     0                 28m
self-awareness-powerconsumptionamd64-v9v7b          1/1     Running     2 (5m22s ago)     28m
self-awareness-powerconsumptionamd64-vt94k          1/1     Running     2 (5m24s ago)     28m
self-awareness-powerconsumptionamd64-wznpb          1/1     Running     2 (5m26s ago)     28m
self-awareness-powerconsumptionamd64-x45f5          1/1     Running     2 (5m22s ago)     28m
self-awareness-powerconsumptionamd64-xjrhd          1/1     Running     2 (5m20s ago)     28m
self-awareness-powerconsumptionamd64-z544c          1/1     Running     0                 27m
self-orchestrator-orchestrator-2nlzk                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-4tt6m                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-69chd                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-6q8s2                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-7htr8                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-7sjfc                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-7x6gq                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-9dgtr                1/1     Running     4 (30h ago)       127d
self-orchestrator-orchestrator-9gkj2                1/1     Running     4 (30h ago)       127d
self-orchestrator-orchestrator-9qhdv                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-9shtf                1/1     Running     7 (31h ago)       129d
self-orchestrator-orchestrator-g6g2r                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-g8zdm                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-gkn7f                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-gv72r                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-gwqhm                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-hpx72                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-hrhtz                1/1     Running     1 (30h ago)       31h
self-orchestrator-orchestrator-jp8np                1/1     Running     1 (30h ago)       31h
```

```
self-orchestrator-orchestrator-k28ph            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-kc4wt            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-kh9dg            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-lrkzf            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-mhtgn            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-mvxl4            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-n9hs2            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-ng8ll            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-phmb9            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-pp4n4            1/1     Running     4 (30h ago)     127d
self-orchestrator-orchestrator-qcdw2            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-r7m52            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-rg6pv            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-vp2xq            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-xhf9n            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-xwzh7            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-z8b2b            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-zb669            1/1     Running     1 (30h ago)     31h
self-orchestrator-orchestrator-zhvzm            1/1     Running     1 (30h ago)     31h
self-security-2rfqn                             4/4     Running     4 (30h ago)     31h
self-security-2xm2t                             4/4     Running     4 (30h ago)     31h
self-security-5bdhc                             4/4     Running     4 (30h ago)     31h
self-security-6rgz6                             4/4     Running     4 (30h ago)     31h
self-security-6s7qx                             4/4     Running     4 (30h ago)     31h
self-security-7scs6                             4/4     Running     4 (30h ago)     31h
self-security-7sklk                             4/4     Running     4 (30h ago)     31h
self-security-8z7fs                             4/4     Running     4 (30h ago)     31h
self-security-9bnnl                             4/4     Running     4 (30h ago)     31h
self-security-bdn8k                             4/4     Running     4 (30h ago)     31h
self-security-bgp8m                             4/4     Running     4 (30h ago)     31h
self-security-bh8ff                             4/4     Running     4 (30h ago)     31h
self-security-c79st                             4/4     Running     4 (30h ago)     31h
self-security-chjsb                             4/4     Running     4 (30h ago)     31h
self-security-dv825                             4/4     Running     4 (30h ago)     31h
self-security-f8sv4                             4/4     Running     12 (30h ago)    127d
self-security-fgj9g                             4/4     Running     4 (30h ago)     31h
self-security-gh94q                             4/4     Running     4 (30h ago)     31h
self-security-gkbkj                             4/4     Running     4 (30h ago)     31h
self-security-hgbbh                             4/4     Running     12 (30h ago)    127d
self-security-j8rjw                             4/4     Running     4 (30h ago)     31h
self-security-jntk4                             4/4     Running     4 (30h ago)     31h
self-security-k4bld                             4/4     Running     4 (30h ago)     31h
self-security-ktchx                             4/4     Running     4 (30h ago)     31h
self-security-l4cv8                             4/4     Running     16 (31h ago)    129d
self-security-mzx5k                             4/4     Running     4 (30h ago)     31h
self-security-nccgh                             4/4     Running     4 (30h ago)     31h
self-security-nkc89                             4/4     Running     4 (30h ago)     31h
self-security-q8l44                             4/4     Running     4 (30h ago)     31h
self-security-shksz                             4/4     Running     4 (30h ago)     31h
self-security-t28qb                             4/4     Running     4 (30h ago)     31h
self-security-v249w                             4/4     Running     4 (30h ago)     31h
self-security-wl42z                             4/4     Running     4 (30h ago)     31h
self-security-xkbsf                             4/4     Running     13 (30h ago)    127d
self-security-xwmsd                             4/4     Running     4 (30h ago)     31h
self-security-zb25q                             4/4     Running     4 (30h ago)     31h
self-security-zpbsx                             4/4     Running     4 (30h ago)     31h
self-security-zt8wr                             4/4     Running     4 (30h ago)     31h
trustmanager-5b9b8855b4-h4lmv                   1/1     Running     0               116m
kklimaszewski:~$ kubectl get pods -n aeros-llo-k8s-system
NAME                                               READY   STATUS    RESTARTS       AGE
llo-k8s-operator-controllermanager-58fc79cb59-22st4   2/2     Running   41 (29h ago)   7d
```

Figure 128. Pilot 2 aerOS Domain 2

Pilot2 – Business Process 1 – Activity - 17 (P2-BP1-I17): Energy Price microservices integration

A microservice-based system was implemented to ensure the integration of energy price prediction data—as well as other energy-related metrics—into the Orion LD context broker (at the heart of aerOS' Data Fabric and federation mechanism) in a consistent and automated manner.

An energy price prediction service exposes a REST API that is queried every hour. (its configurable) Upon request, the system checks whether relevant market input data has changed since the last query. If so, a new prediction is generated; otherwise, the previously calculated forecast is returned. Regardless of whether the data is new or repeated, it is always pushed to Orion LD in the agreed JSON-LD format, ensuring continuous data availability and synchronization.

The solution architecture includes three microservices:

- **tgescrapper**, which collects current market data,
- **price_prediction**, which hosts the prediction model and provides API access to forecasts,
- **data_connector**, which queries the prediction API and forwards the output to Orion LD.

Modules *tgescrapper* and *price_prediction* share a common database, enabling seamless internal communication. In addition to price predictions, the *data_connector* also gathers and transmits contextual information from the EMACS APIs and its supporting services—such as load, power production, and weather data.

Pilot2 – Business Process 1 – Activity - 18 (P2-BP1-I18): Container deployment in new location

The container with two edge nodes (physically separated) is moved to the final location, which is Łódź, Poland. This was properly reported in the first amendment to the Grant Agreement of aerOS. This change has not affected pilot's activities, scope or ambition.

To relocate the container with the hardware, the following tasks were carried out: disassembly of the hardware, draining of the chiller system, transport of the container to the designated site, and reassembly of all components.

Figure 129. Pilot 2 Container and chiller relocation to Łódź, CF location

Pilot2 – Business Process 1 – Activity - 19 (P2-BP1-I19): Container connection to PV (second green energy source)

Figure 130. Pilot 2 Container connection to PV

After relocating the container to its final location, one of the edge nodes was connected to a dedicated photovoltaic (PV) installation. **The aim is to power the system with renewable energy and to enable monitoring of energy consumption, therefore allowing to achieve the innovations proposed in the pilot.**

In order to do so, a market analysis of available PV providers was conducted, evaluated offers based on technical parameters (e.g., panel efficiency, inverter type), and selected the most cost-effective and technically suitable solution. A contract was signed for the delivery and installation of the PV system on the container's roof.

The installed PV system consists of 9 high-efficiency monocrystalline panels with a total capacity of 2 kWp, connected to a inverter. The inverter supports both LAN and Wi-Fi connectivity and is equipped with a Modbus RTU interface, enabling seamless integration with external monitoring and control systems. This communication interface allows the collection of detailed real-time data such as power output, voltage, current, and inverter status.

The connection between the edge node and PV installation was set up in the foreseen time.

Pilot2 – Business Process 1 – Activity - 20 (P2-BP1-I20): aerOS Basic components (up-to -date version)

All the basic components are updated on Pilot 2 infrastructure to their up-to-date version as of July 2025. Versions of the components present there are displayed in table 1 in section 2. Some of the components do not maintain a proper versioning scheme (ex. HLO components) so image hashes were used to pin them to use latest versions available on specified dates. These actions were agreed with the technical development team of aerOS.

Some components presented dependencies with internal components (e.g., ldap-collector), and some others (e.g, AyncAPI, self-configuration) were not suited to accomplish the goals of the pilot, therefore were not deployed. Note here that these were not basic -but auxiliary- components of aerOS.

Currently Pilot 2's installation of aerOS is running, able to schedule workloads on two containerized edge nodes and is available to logged in users at https://portal.pilot2.aeros-project.eu/ .

Pilot2 – Business Process 1 – Activity - 21 (P2-BP1-I21): Integration in the DF the next data sources "The data sources include energy meters, weather sensors and data concentrators that collect multiple power parameters from all electronics devices installed at given facility"

After connecting all meters to the CloudFerro installation and the Electrum SCADA system, Pilot 2 is able to retrieve data, including from energy meters, weather sensors, and data concentrators that aggregate various power parameters from all electronic devices deployed at the facility.

Data transmitted from ELEC components to the Data Fabric is prepared in the NGSI-LD format, with custom entities. The entities provide a schema for pilot-specific parameters (such as weather predictions from the weather station), and re-use schemas from the aerOS ontology (e.g. Infrastructure Element) to connect the new information from the pilot, to information managed by core aerOS.


```json
{
    "id": "urn:ngsi-ld:PricePredictions:PL-PricePredictions-ELEC-2025-04-03-123",
    "type": "PricePredictions",
    "dateIssued": "2025-04-01T11:30:00.004033",
    "https://w3id.org/aerOS/power#predictions": [
        {
            "validFrom": "2025-04-03T00:00:00",
            "validTo": "2025-04-03T10:00:00",
            "https://w3id.org/aerOS/power#f1_price": 445.1985168457031
        },
        {
            "validFrom": "2025-04-03T01:00:00",
            "validTo": "2025-04-03T02:00:00",
            "https://w3id.org/aerOS/power#f1_price": 416.38592529296875
        },
        {
            "validFrom": "2025-04-03T02:00:00",
            "validTo": "2025-04-03T03:00:00",
            "https://w3id.org/aerOS/power#f1_price": 393.5117492675781
        }
    ],
    "@context": [
        "https://fiware.github.io/data-models/context.jsonld",
        "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
    ]
}
```


Figure 131. Price Predictions NGSI-LD with multiple predictions in a single entity.

```json
{
  "id":"urn:ngsi-ld:VirtualPowerSource:001",
  "type":[
    "Device",
    "a4e:PowerSource",
    "a4e:VirtualPowerSource"
  ],
  "description":{
    "type":"Property",
    "value":"Current farm power production and forecast"
  },
  "location":{
    "type":"GeoProperty",
    "value":{
      "type":"Point",
      "coordinates":[
        21.0122,
        52.2297
      ]
    }
  },
  "https://w3id.org/aerOS/power#greenEnergyRatio": 1.0,
  "https://w3id.org/aerOS/power#:isStable":false,
  "https://w3id.org/aerOS/power#power": 23.25518,

  "https://w3id.org/aerOS/power#predictions": [
      {
          "validFrom": "2025-04-03T06:00:00",
          "validTo": "2025-04-03T07:00:00",
          "https://w3id.org/aerOS/power#power": 31.80207061767578
      },
      {
          "validFrom": "2025-04-03T07:00:00",
          "validTo": "2025-04-03T08:00:00",
          "https://w3id.org/aerOS/power#f1_price": 91.32544708251953
      },
      {
          "validFrom": "2025-04-03T08:00:00",
          "validTo": "2025-04-03T09:00:00",
          "https://w3id.org/aerOS/power#f1_price":133.32247924804688
      },
      {
          "validFrom": "2025-04-03T09:00:00",
          "validTo": "2025-04-03T10:00:00",
          "https://w3id.org/aerOS/power#f1_price":96.40310668945312
      },
      {
          "validFrom": "2025-04-03T10:00:00",
          "validTo": "2025-04-03T11:00:00",
          "https://w3id.org/aerOS/power#f1_price":69.22946166992188
      }
   ],
  "@context": [
      "https://fiware.github.io/data-models/context.jsonld",
      "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
   ]
}
```


Figure 132. Power Source with maximum power output potential predictions.

```json
{
    "id": "urn:ngsi-ld:WeatherForecast:PL-WeatherForecast-Mazovia-2025-03-31",
    "type": "WeatherForecast",
    "refDevice": "urn:ngsi-ld:Device:WeatherStation:Electrum-AerOS-12345",
    "address": {
        "type": "Property",
        "value": {
            "addressLocality": "Mazovia",
            "addressCountry": "PL"
        }
    },
    "location": {
        "type": "GeoProperty",
        "value": {
            "type": "Point",
            "coordinates": [
                21.0122,
                52.2297
            ]
        }
    },

    "dateRetrieved": "2025-03-31T09:29:45.937",
    "dateIssued": "2025-03-31T09:29:45.937",

    "validFrom": "2025-03-31T09:29:45.937",
    "validTo": "2025-04-1T19:29:45.937",

    "temperature": 7.5,
    "dewPoint": 7.099996,
    "windSpeed": 1.374,
    "windDirection": 321.2818,
    "atmosphericPressure": 990.7993,
    "relativeHumidity": 97.5,
    "visibility": 0.0,
    "solarRadiation": 174,
    "https://w3id.org/aerOS/weather#clouds": {
        "type": "Property",
        "value": 0.0
    },
    "https://w3id.org/aerOS/weather#gni": {
        "type": "Property",
        "value": 0.0
    },
    "https://w3id.org/aerOS/weather#ghi": {
        "type": "Property",
        "value": 10.0
    },
    "https://w3id.org/aerOS/weather#dni": {
        "type": "Property",
        "value": 0.0
    },
    "https://w3id.org/aerOS/weather#dhi": {
        "type": "Property",
        "value": 10.3
    },
    "@context": [
        "https://smart-data-models.github.io/dataModel.Weather/context.jsonld",
        "https://raw.githubusercontent.com/smart-data-models/dataModel.Weather/master/context.jsonld",
        "https://fiware.github.io/data-models/context.jsonld",
        "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
    ]
}
```


Figure 133. Weather Prediction NGSI-LD entity.

```json
{
    "id": "urn:ngsi-ld:WeatherObserved:PL-WeatherObserved-Mazovia-2025-03-31",
    "type": "WeatherObserved",
    "refDevice": "urn:ngsi-ld:Device:WeatherStation:Electrum-AerOS-12345",
    "address": {
        "type": "Property",
        "value": {
            "addressLocality": "Mazovia",
            "addressCountry": "PL"
        }
    },
    "location": {
        "type": "GeoProperty",
        "value": {
            "type": "Point",
            "coordinates": [
                21.0122,
                52.2297
            ]
        }
    },
    "dateObserved": "2025-03-31T09:29:45.937",
    "temperature": 7.5,
    "dewPoint": 7.099996,
    "windSpeed": 1.374,
    "windDirection": 321.2818,
    "atmosphericPressure": 990.7993,
    "relativeHumidity": 97.5,
    "visibility": 0.0,
    "solarRadiation": 174,
    "https://w3id.org/aerOS/weather#clouds": {
        "type": "Property",
        "value": 0.0
    },
    "https://w3id.org/aerOS/weather#gni": {
        "type": "Property",
        "value": 0.0
    },
    "https://w3id.org/aerOS/weather#ghi": {
        "type": "Property",
        "value": 10.0
    },
    "https://w3id.org/aerOS/weather#dni": {
        "type": "Property",
        "value": 0.0
    },
    "https://w3id.org/aerOS/weather#dhi": {
        "type": "Property",
        "value": 10.3
    },
    "@context": [
        "https://smart-data-models.github.io/dataModel.Weather/context.jsonld",
        "https://raw.githubusercontent.com/smart-data-models/dataModel.Weather/master/context.jsonld",
        "https://fiware.github.io/data-models/context.jsonld",
        "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
    ]
}
```


Figure 134. Weather Observed NGSI-LD entity.

2.2.2. Secure Federation of edge/cloud

2.2.2.1. Technical Schema

Scenario 2 has adopted an approach focused on the orchestration of mixed workloads from multiple users (tenants) over an aerOS continuum. The main reason behind this approach is that a cross-cloud edge set up for the whole duration of the project based on rented digital infrastructure would imply a high cost (of the Internet egress (e.g. GPC ~0,12$/GiB[20]) and the need of provisioning Kubernetes cluster), while pushing away the real goal of the demonstration of the pilot.

Therefore, it was decided to enhance Earth Observation processing data with different workloads (like text processing, less intense ML inference/training, also World Community Grid [21]is being used).

It better shows strengths of the aerOS ecosystem than sending TBs of data back and forth via Internet just for the processing.

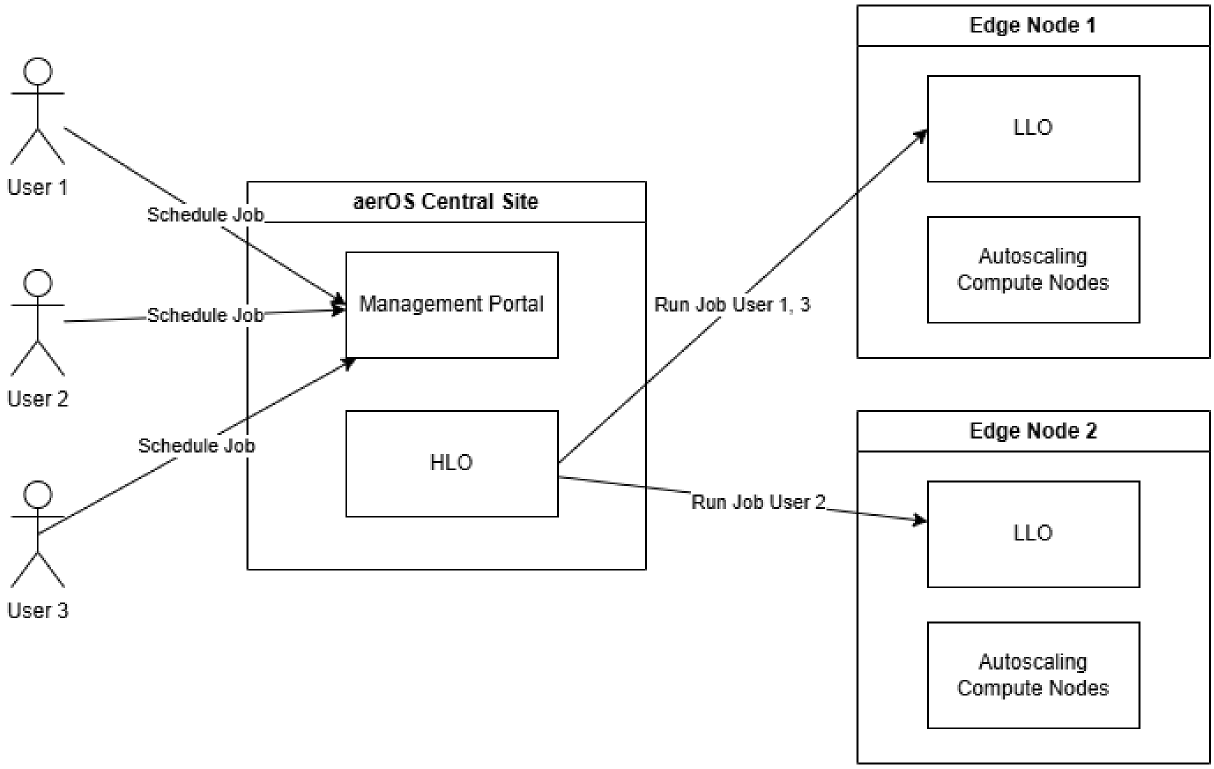


Figure 135. Pilot 2 Secure Federation of edge/cloud Ontology diagram

In the Scenario 2 overview can be seen that different users can schedule their own processing tasks with different parameters and different docker image. And for this scenario it is important not to restrict in advance the type of jobs or tasks. Therefore, there is no need to describe them. Tasks can be adjusted during the Scenario to observe the changes in aerOS behaviour depending on the workload.

2.2.2.1. Report of activities

2.2.2.1.1. Setup and procurement activities

Pilot2 – Business Process 2 – Activity - 3 (P2-BP2-SA3): Preparation for untrusted workloads

To prevent aerOS workloads from accessing network resources of other workloads and CloudFerro's internal resources, cillium network policies are put into place in default namespace of all clusters in pilot 2's infrastructure. These policies restrict egress to the internet, DNS server, krakend gateway and aerOS services.

There were plans to use kata containers to further isolate workloads from hosts they are running on. However, to use kata-containers one needs to specify runtimeClassName field on deployed pods. At the current version of aerOS, the component LLO does not allow specifying it on created pods and there is no way to set a default

[20] https://cloud.google.com/vpc/pricing?hl=es#internet_egress
[21] https://www.worldcommunitygrid.org

runtime class for namespace. This means that although using kata containers is possible on our clusters, aerOS cannot be configured to use them.

This is expectable, since it was not the goal of the project to cover kata containers compliance.

There is no need to consider additional development or integration activities, since this action allows for achieving a secure federation of edge-cloud schema in Pilot 2.

2.2.3. Pilot 2 Time-plan

The following Gantt diagram list all the activities from M19 (February 2024) until M35 (July 2025), representing evolution from status declared in D5.3.

Table 7. Pilot 2 updated Gantt timeline

Pilot 2		2024											2025								
Code	**Name**	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33	M34	M35	M36	M37	M38
Business Process 1 (Scenario 1) - Green Edge Processing																					
Setup & Procurement Activities																					
P2-BP1-SA1	A1- Obtaining servers and switches																				
P2-BP1-SA2	A2 - Obtaining RACKs, servers and switches for second container																				
P2-BP1-SA3	A3 - Preparation for untrusted workloads																				
Development Activities																					
P2-BP1-DA4	A4 - HW installation and run test in container																				
P2-BP1-DA5	A5 - K8s setup and test																				
P2-BP1-DA6	A6 - HW installation and run test in the LAB																				
P2-BP1-DA7	A7 - Containerized Edge Node Integration with Electrum components																				
P2-BP1-DA8	A8 - Containerized Edge Node Integration with SRIPAS components																				
P2-BP1-DA9	A9 - Lab Edge Node Integration with aerOS components																				
P2-BP1-DA10	A10 - Autoscaler monitor Development																				
P2-BP1-DA11	A11 - Development of future Energy Price microservices																				
P2-BP1-DA12	A12 - Development of DF the data sources connectors																				

Integration Activities																					
P2-BP2-IA10	A10 - First Containerized Edge Node Integration with Electrum components																				
P2-BP2-IA11	A11 - First Containerized Edge Node Integration with SRIPAS components																				
P2-BP2-IA12	A12 - Container deployment																				
P2-BP2-IA13	A13 - Container connection to PV																				
P2-BP2-IA15	A15- aerOS Basic components																				
P2-BP2-IA16	A16 - aerOS Non Basic components																				
P2-BP2-IA17	A17 - Energy Price microservices integration																				
P2-BP2-IA18	A18 - Container deployment in new location																				
P2-BP2-IA19	A19 - Container connection to PV (second green energy source)																				
P2-BP2-IA20	A20 - aerOS Basic components(up-to-date version)																				
P2-BP2-IA21	A21 - Integration in the DF the next data sources																				
Validation Activities																					
P2-BP2-VA17	A17 - First Containerized Edge Node test																				
P2-BP2-VA18	A18 - K8s setup and test																				
P2-BP2-VA19	A19 - Second Containerized Edge Node test																				
P2-BP2-VA20	A20 - Both Containerized Edge Node run test with aerOS																				
P2-BP2-VA21	A21 - HW installation and run test in container																				
P2-BP2-VA22	A22 - Scenario 1 deployment and test																				
P2-BP2-VA23	A23 - Scenario 1 lessons learned																				

P2-BP2-VA27	A27 - KPI validation (1st version)																				
P2-BP2-VA28	A28 - KPI validation (final)																				
Business Process 2 (Scenario 2) - Secure Federation of edge/cloud																					
Setup & Procurement Activities																					
Development Activities																					
Integration Activities																					
P2-BP2-IA14	A14 - Inter-cloud integration																				
Validation Activities																					
P2-BP2-VA24	A24 - Configuration Validation test																				
P2-BP2-VA25	A25 - Scenario 2 deployment and test																				
P2-BP2-VA26	A26 - Scenario 2 lessons learned																				

2.3. Pilot 3 - High Performance Computing Platform for Connected and Cooperative Mobile Machinery to improve CO2 footprint

Pilot 3 has undergone a revision in its implementation timeline due to the seasonal constraints of agricultural operations. Specifically, the **field validation activities** have been rescheduled to take place in **August and September**, aligning with the operational window of the demonstration field. This adjustment was necessary because certain critical activities in the field can only be performed during this period.

The key activities dependent on this timeframe include:

- Setup and calibration of the autonomous sprayer system
- Execution of AI-based weed detection under real field conditions
- Validation of the georeferenced application map through live spraying trials

As a result, the following pilot actions have been **postponed**:

- **Field validation of the AI model utilizing the aerOS components and spraying system**
- **Performance assessment under operational conditions**
- **Final aerOS integration testing with physical machinery**

Despite this delay, all other activities that are independent of physical field availability have been successfully completed. These include:

- Integration of the aerOS framework in the Lab setup including testing
- Development and training of the AI weed detection model
- Integration of telemetry and GNSS data into the SESAM 2 system
- Configuration of the communication and cloud infrastructure (e.g., MECSware campusXG setup)

This phased approach ensures that the project remains on track, with technical components fully prepared for deployment once field access becomes feasible.

In preparation for the development and integration activities of the pilot, several foundational steps were taken to ensure the success of Pilot 3. TTControl and John Deere collaborated to integrate the High Performance Computing Platform (HPCP) into prototype vehicles, including the SESAM 2 electric tractors. Hardware components such as GNSS receivers, UE modems, and high-resolution cameras were installed and tested for connectivity and data acquisition. Embedded software on both companies' platforms was adapted to support future integration with aerOS components. **Preliminary AI applications for spraying optimization and row detection were tested using internal datasets**. Middleware and containerized execution concepts were prepared, along with APIs and data formats. A laboratory setup was established to simulate the edge-cloud continuum and validate early KPIs. Integration workshops and remote sessions facilitated hardware and software alignment.

From the perspective of the **1st Scenario (Cooperative large-scale production)**

The primary goal of this scenario has been to assess the current fieldwork operations to identify areas for improvement. By leveraging the advanced Electronic Control Unit (ECU) from TTControl, John Deere aims to enhance its fieldwork processes. The computing capability of the ECU serves as a catalyst for optimizing performance and efficiency within agriculture and construction tasks, enabling significant advancements in these areas.

Essential tasks in agriculture and construction often necessitate the collection and analysis of data from various components. The integration of a High-Performance ECU facilitates both on-board and off-board data processing, significantly impacting the overall analysis of fieldwork.1

Pilot scenario 1 focuses on improving John Deere's fieldwork operations using the prototypes developed in the project to achieve substantial benefits in the overall performance. This innovative approach ensures a continuous drive towards elevated efficiency in field operations and becomes a benchmark for future advancements in the industry.2

Regarding the **2nd Scenario (Basis for CO_2 neutral Intelligent Operation):**

The foundation for CO_2-neutral intelligent operations, including farming, construction, and forestry, relies among others on edge computing technology. By deploying tasks at the edge, one can drastically reduce latency and reaction time, especially when utilizing low-latency networks such as 4G or 5G. This innovative approach paves the way for more efficient and sustainable operations that minimize environmental impact while maximizing productivity.

John Deere electrified vehicles are the only way forward to reduce the CO_2 footprint in the farming or other domains because the energy-saving aspect directly contributes to lowering CO_2 emissions, thereby minimizing the ecological footprint of various industries. By usage of collaborative swarm vehicle scenarios, even further potential energy reductions will be possible. The operations and corresponding operation domains are presented in the following Figure .

In addition, the integration of AI-supported strategies has proven to greatly enhance overall performance while adhering to sustainable and environmentally conscious practices. By leveraging the power of edge computing and low-latency networks, as well as applying the AI-supported and data-driven applications, agricultural companies can work towards achieving CO_2-neutral future for our planet. With these advanced solutions in place, organizations can simultaneously meet demands for higher efficiency while addressing the pressing need for environmental responsibility.

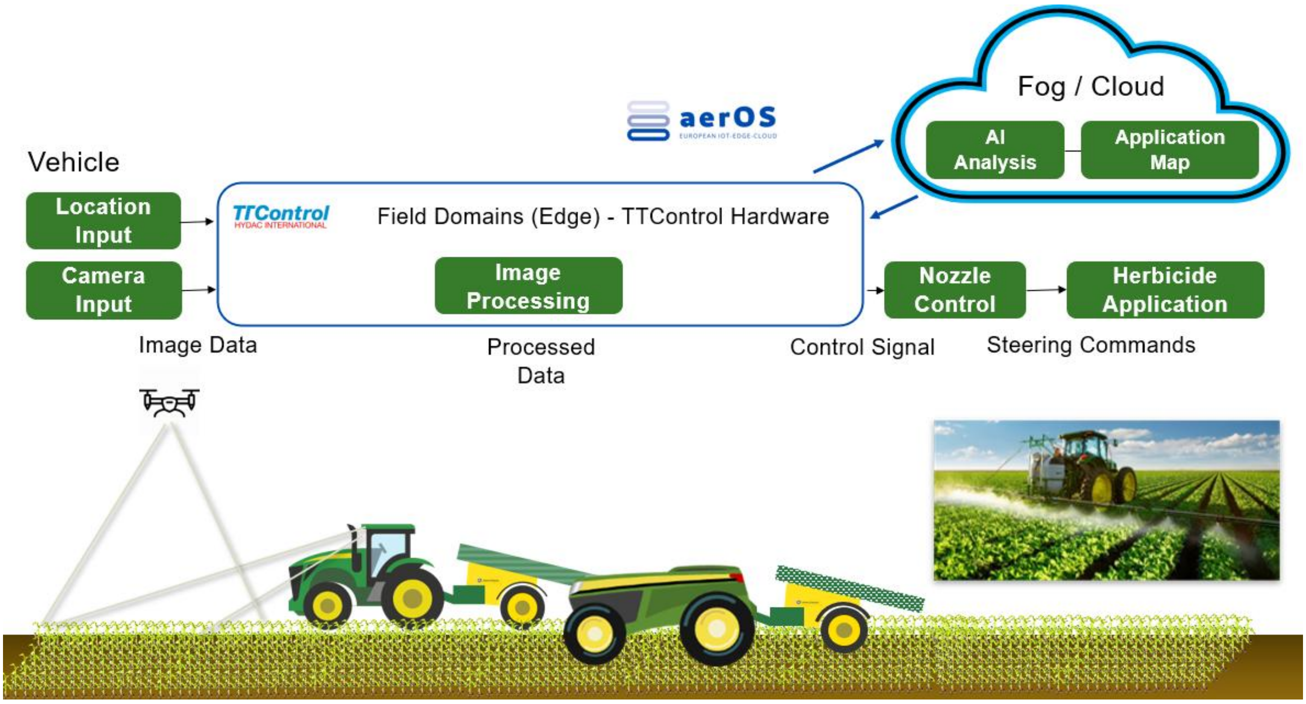


Figure 136. System setup for scenario 1 and 2

The following table showcases how Pilot 3 has switched from processing data in the Cloud to the Edge.

Table 8. Pilot 3 Edge vs Cloud comparison before and after aerOS.

	Cloud	Edge
Before aerOS	No Edge-Cloud processing available.	Standalone execution of AI component.

After aerOS	Distributed execution of AI component into the cloud, leveraging cloud scale intelligence (for training models) and swarm distribution of optimized maps	Distributed execution of AI component lead to decreasing processing load from the vehicle processing unit and enable scalability for the AI execution

2.3.1. Cooperative large-scale production (Scenario 1)

2.3.1.1. Technical schema

Pilot 3 operates within two aerOS Domains, as illustrated in Figure :

Domain 01 is the pilot's entry domain. It includes all aerOS services required for the administration of the pilot continuum and comprises two infrastructure elements: the Master Node and the Edge Node.

- The Master Node acts as head of domain, and hosts the main components for the aerOS Met-aOS continuum establishment and maintenance
- The Edge Node handles video compression in preparation for the AI application (that later can run in the cloud).
- Communication between the nodes is facilitated via KubeEdge MQTT.

Domain02 is the cloud domain, responsible for executing the AI application for weed detection and generating a spatially precise prescription map.

- The connection between the field and the cloud domain is established via HTTP over Wire-Guard VPN.

The generated prescription map will be distributed to multiple in-field machines for coordinated herbicide application (see Figure).

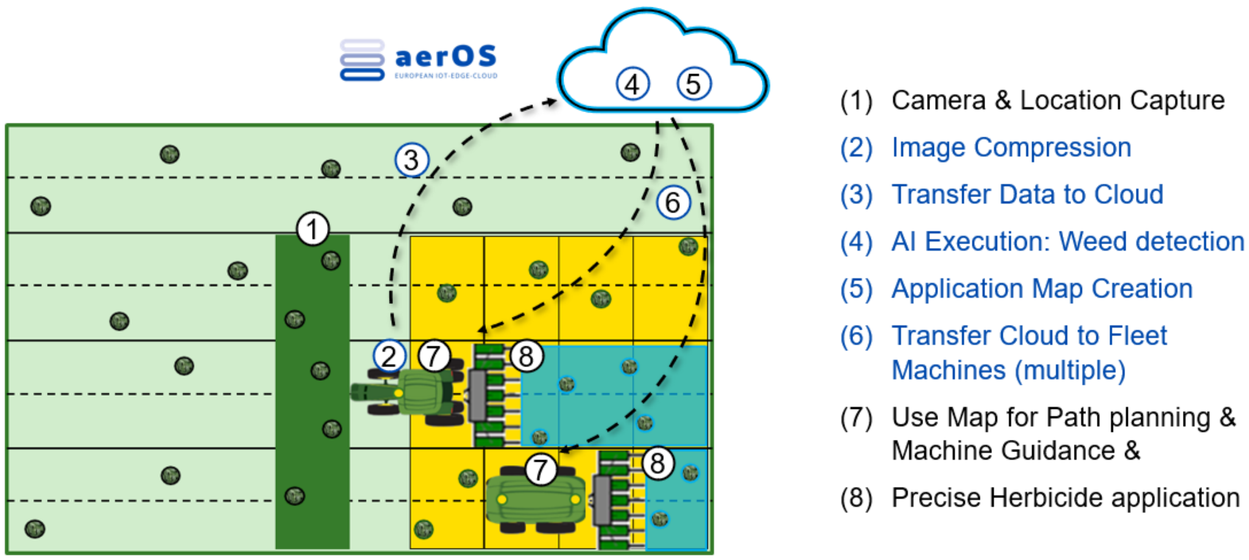


Figure 137. Cooperative machine coordination

In view of an adequate testbed required for KPI evaluation, Figure 137 illustrates a comprehensive schematic of the component topology within a corresponding testbed system. While the depicted evaluation support infrastructure is specifically used to establish a hardware-in-the-loop (HiL) testing system in an isolated laboratory environment, its generic architecture can be adapted to support in-field evaluation of Pilot 3 in the same manner. Therefore, the inter-node communication links based on Ethernet have to be exchanged by 5G technology, while the service container registry can be moved to a publicly available repository with secure data access support.

The applied evaluation setup based on that testbed architecture is used as basis for systematic evaluation of the aerOS use case in Pilot 3 against selected key performance indicators including vehicle performance improvement (KPI 2.3.2) and emission reduction (KPI 2.3.3). The diagram in Figure 138 details the deployment of aerOS basic and additional components including **the use case-specific AI application service of John Deere, which are orchestrated among two aerOS domains**. These domains include the Entry/Field Domain hosting a KubeEdge-based Kubernetes cluster and the Cloud Domain, which hosts a K8s-based Kubernetes cluster. Both service clusters comprise the required basic and additional aerOS components to implement a distributed service continuum featuring vertical scaling in terms of potentially multiple edge and cloud nodes as well as horizontal scaling with respect to potentially further service clusters.

For the purpose of performing concrete measurement trials to evaluate the KPIs in question, the HiL testbed setup consists of an edge computing node running on the TTControl ECU in conjunction with a single Cloud service instance running on the high-performance workstation of John Deere. Both testbed components are instantiated via a multi-purpose machine, IESE Laptop, which is dedicated for service deployment and administration via the aerOS Management Portal Web application. This testbed is complemented with further infrastructure devices supporting the deployment and evaluation of the aerOS use case by providing a local container registry for hosting the IP-protected AI service of John Deere along with public Internet access for pulling official aerOS component updates.

This testbed instantiation allows for focused observation of individual metrics, which have been identified as relevant to substantiate the aspired KPIs in question. More details on the individual testbed components including technical specifications and development activities are given in the following subsections.

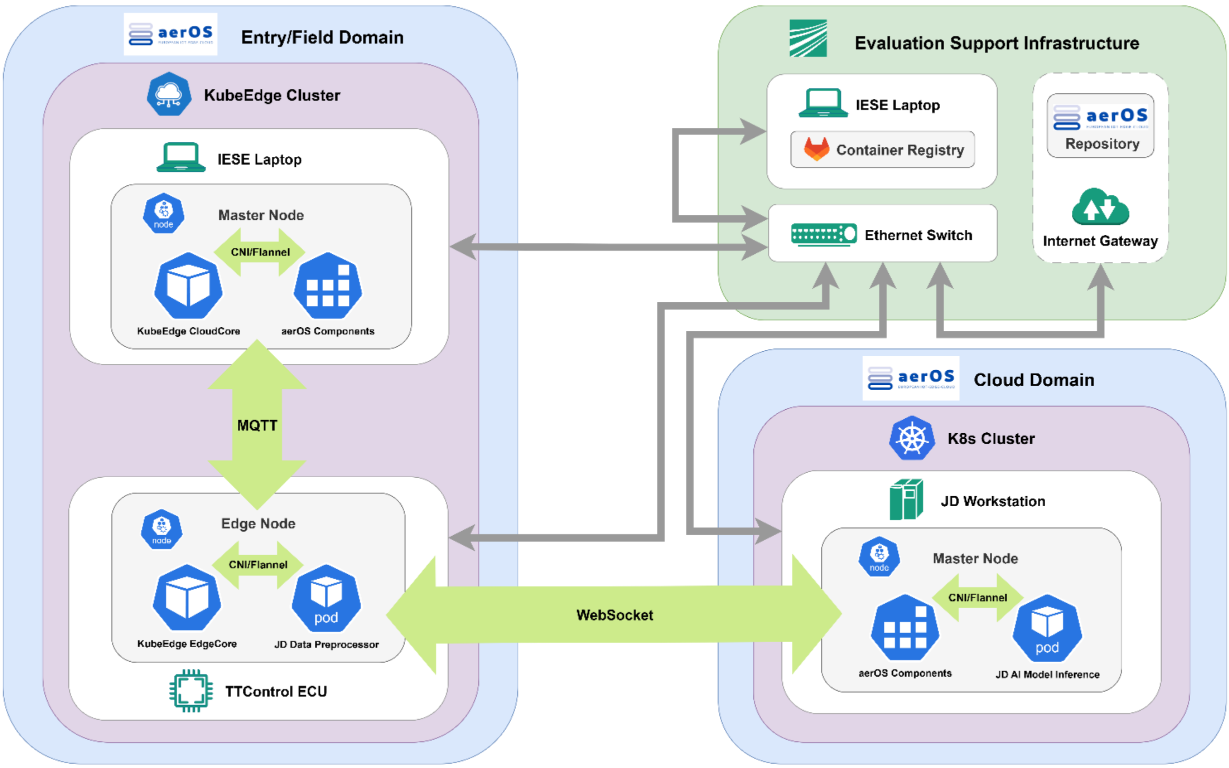


Figure 138. Deployment diagram of aerOS and supporting infrastructure components describing the evaluation testbed setup of the aerOS-based Pilot 3 use case for scenarios 1 and 2.

2.3.1.2. Report of activities

2.3.1.2.1. Setup and procurement activities

Pilot3 - BusinessProcess1 - SetupActivity-1 (P3-BP1-SA1) Procurement Ethernet-based ECU platform prototype (TTC):

The TTC Motion prototype board has been equipped with the NVIDIA Jetson AGX Xavier system, a high-performance embedded computing platform designed for AI workloads. The software stack is based on the JetPack 4.2.2 SDK, which includes:

- Linux kernel and file system tailored for Jetson Xavier.
- CUDA libraries for GPU-accelerated computing.
- TensorRT for optimized deep learning inference.
- GStreamer and Multimedia API for real-time video processing.
- TensorFlow for model development and deployment.
- AR0231 camera drivers integrated for high-resolution image capture.

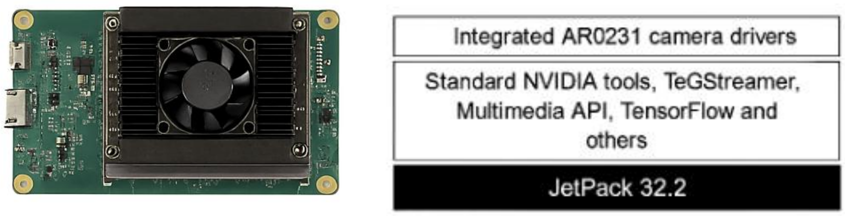


Figure 139. TTC Motion prototype board with Jeston Xavier Module + schematic

Pilot3-BusinessProcess 1-SetupActivity-2 (P3-BP1-SA2) Setup Sesam 2:

For the in-field execution of Pilot 3, the setup of the John Deere Sesam 2 electrical tractor was executed. The John Deere Sesam 2 is an 8.5 metric ton automated electric tractor with 1,000 kWh of energy storage and 500 kW (680 hp) of power. It comes with a 1-kilometer-long cable (max 2.5 kV/300 kW) which rolls on and off automatically. The electric power unit delivers 100 kW (136 HP) to the wheels, and 200 kW (272 HP) to power additional machinery.

In order to prepare the John Deere Sesam 2 Tractor for testing, technicians had fully charged the battery, conducted system diagnostics, and calibrated the electric motors and control systems. Safety checks and software updates were also completed prior to field deployment.

Figure 140. John Deere Sesam 2 Tractor

Pilot3- BusinessProcess1-SetupActivity-3 (P3-BP1-SA3) Setup Sprayer:

Pilot 3 has undergone a revision in its implementation timeline due to the seasonal constraints of agricultural operations. Specifically, the **field validation activities** have been rescheduled to take place in **August and September**, aligning with the operational window of the demonstration field. For the Sprayer setup a 6215R tractor serves as a power and control platform for pulled sprayers like the R975i.

It integrates with ExactApply™ and John Deere Operations Center™ for precision spraying. Uses Gen 4 CommandCenter™ display and AutoTrac™ with StarFire™ RTK for sub-inch accuracy.

The aerOS components described in the testbed setup are used on the platform as well.

Figure 141. John Deere 6215R Tractor.

Figure 142. John Deere R975i Sprayer

Table 9. Technical specification of John Deere 6215R Tractor.

Specification	Value
Engine	6.8L 6-cylinder PowerTech™ PSS diesel
Gross Power	215 hp (160.3 kW)
Max Power	252 hp (187.9 kW)
PTO Power	177 hp (claimed), 205.72 hp (tested)
Transmission Options	16-speed, 20-speed, 24-speed power shift, and IVT
Hydraulics	Closed center pressure flow compensated (PFC), 2900 psi, 41 gpm flow
3-Point Hitch	Rear Type III/IIIN, Lift Capacity 12,015 lbs; Front Hitch IIIN, Lift Capacity 7275 lbs
Fuel Capacity	104.4 gal (395.2 L)
Weight	~19,610 lbs (8895 kg)
Cab	ComfortView™ with active hydraulic suspension and AutoTrac™ optional
PTO Options	Multiple configurations including 540/1000 and 540E/1000E

Table 10. Technical specification of John Deere R975i Sprayer.

Specification Parameter	Value
Working Width	24 meters
Transport Dimensions	Length 7.63 m, Width 3.00 m, Height 3.65 m
Tank Volume	7500 liters
Pump Type	PowrSpray dual-circuit system
Pump Capacity	560 l/min (filling), up to 1200 l/min (spraying)
Tyres	380/90 R46
Weight	5410 kg
Boom Folding	2-fold hydraulic
Partial Widths	8 sections
Slope Control	Hydraulic hillside/slope control available
Nozzle Spacing	Optional 25 cm

Application Accuracy	Up to 98% with Direct Rate Control

Pilot3- BusinessProcess1-SetupActivity-4 (P3-BP1-SA4) Setup StarFire GNSS Receiver:

The sensor used is the John Deere **StarFire™ 7500 Receiver**. The StarFire™ 7500 Receiver is a high-precision GNSS receiver designed for advanced agricultural guidance and automation. It supports multiple satellite constellations and offers RTK-like accuracy without requiring additional hardware. The receiver is ideal for operations such as planting, spraying, and harvesting, and integrates seamlessly with John Deere's

Figure 143. The StarFire™ 7500 Receiver

AutoTrac™ systems. See the table below for detailed specification.

Table 11. Specification StarFire™ 7500 Receiver.

Feature	Specification
Correction Signal Options	SF1, SF-RTK, Radio RTK, Mobile RTK
Accuracy (SF-RTK)	±2.5 cm horizontal pass-to-pass
Accuracy (SF1)	±15 cm horizontal pass-to-pass
Satellite Constellations	GPS, GLONASS, Galileo, BeiDou
Pull-In Time (SF-RTK)	Up to 73% faster than SF3 (approx. 3× faster than StarFire 6000)
Repeatability	Up to 5 years season-to-season (SF-RTK)
Terrain Compensation Module (TCM)	Integrated; adjusts for roll, pitch, and yaw
Signal Switching	Automatic switching between StarFire satellites for optimal performance
Connectivity	Ethernet (for over-the-air updates), JDLink™ compatible
Security Features	Optional locking bracket for theft protection

Pilot3- BusinessProcess1-SetupActivity-6/7 (P3-BP1-SA6/7) Procurements & Setup Ethernet Camera:

The camera used has a 1 /2.3 inch image sensor with 12 megapixels and the maximum video resolution is 3840*2160 pixels at 25fps in a 16:9 format. The Ethernet camera was set up by connecting it to the network via an Ethernet cable, powering it through PoE, identifying its IP address, accessing its web interface, and configuring the network and video settings.

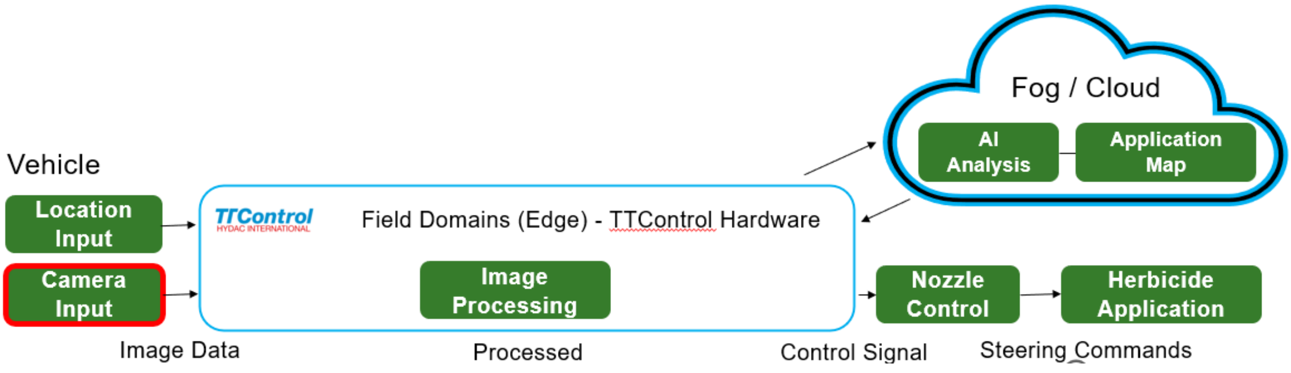


Figure 144. Camera Setup Diagram

Pilot3- BusinessProcess1-SetupActivity-8 (P3-BP1-SA8) Procurement Computing node (on-premise domain):

The compound computing node within the on-premise Entry/Field domain consists of two physical devices denoted as "IESE Laptop" provided by Fraunhofer IESE and "TTControl ECU" provided by TTControl, respectively, as per Figure .

The laptop device features multi-purpose hardware running a publicly available open-source operating system based on the following technical specifications.

Table 12. Technical specification of computing node "IESE Laptop".

Specification Parameter	Value
Model	Lenovo ThinkPad T14s Gen 2i
Processor	Intel Core i7-1165G7
Memory	16 GB
Graphics	Mesa Intel Xe Graphics TGL GT2
Networking	1Gbit/s Ethernet
Storage	512 GB
Operating System	Debian GNU/Linux 12 (bookworm)

The ECU device features dedicated Socket-on-Chip (SoC) hardware running a customized open-source operating system based on the following technical specifications.

Table 13. Technical specification of computing node "TTC Control ECU".

Specification Parameter	Value
Model	Jetson AGX Xavier
Processor	NVIDIA Carmel Arm Version 8.2 with 8 Cores und 64-Bit-CPU 8 MB L2 + 4 MB L3
Memory	32 GB 256-Bit-LPDDR4x
Graphics	NVIDIA Volta-GPU 512 Cores 64 Tensor-Cores 1377 MHz
Networking	1Gbit/s Ethernet
Storage	32 GB eMMC 5.1
Operating System	Ubuntu 18.04 LTS

Pilot3- BusinessProcess1-SetupActivity-9 (P3-BP1-SA9) Setup Computing node (on premise domain):

The computing node was set up within the on-premise Entry/Field Domain to support localized data processing, secure resource management, and seamless integration with internal infrastructure for high-performance computing tasks

The purpose of the laptop device as part of the computing node in question, provided by Fraunhofer IESE is to facilitate deployment and instantiation of aerOS basic and additional components among adjacent service domains and actual edge devices like the ECU device provided by TTControl in the evaluation setup. To this end, the following aerOS components have been deployed (see Figure).

The purpose of the ECU device, in turn, is to capture sensor data, execute image processing routines, and forward data for further interpretation and analysis if required for performance reasons.

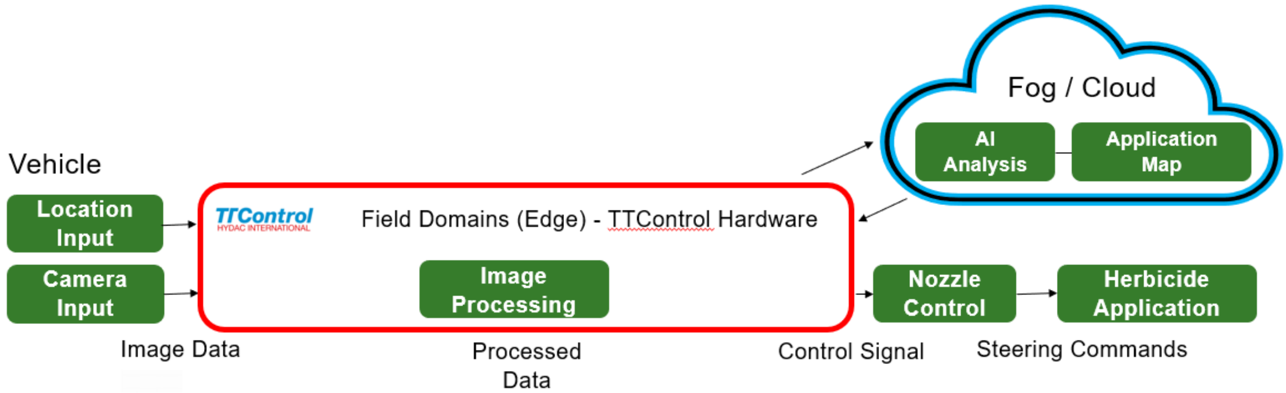


Figure 145. On premise domain

Pilot3- BusinessProcess1-SetupActivity-10 (P3-BP1-SA10) Setup Extended SESAM 2 with consumption analytics (iterative with A2.1.3):

The extended SESAM 2 system was set up with integrated consumption analytics to monitor and evaluate energy usage. This was implemented by enabling machine telemetry to capture electricity consumption (kWh/ha) through the collection of power rating data (kW) and operating time (hours). Additionally, satellite positioning information was used for field mapping to determine the hectares (ha) covered. Those dater are captured in the ECU Field domain.

Pilot3- BusinessProcess1-SetupActivity-11 (P3-BP1-SA11) Setup Adapted Computing Node (iterative with A2.1.11):

The Cloud computing node denoted as "JD Workstation" in Figure 136 represents the aerOS Cloud domain and provides enhanced computing capabilities for AI model inference along with the necessary aerOS services to deploy the computation-intensive counterpart of the two distributed AI service components of John Deere for weed detection and sprayer target optimization.

This high-performance computing (HPC) device runs a publicly available open-source operating system based on the following technical specifications.

Table 14. Technical specification of Cloud computing node "JD Workstation".

Specification Parameter	Value
Model	HP Z8 Fury G5
Processor	Intel Xeon w5-3435X
Memory	128 GB
Graphics	NVIDIA RTX A6000

Networking	1Gbit/s Ethernet
Storage	6 TB
Operating System	Ubuntu 24.04.2 LTS

For the infield communication between the Entry/Field domain and the Cloud domain, a private **5G setup** provided by **MECSware campusXG - 5G-SK-O-4108** is in use.

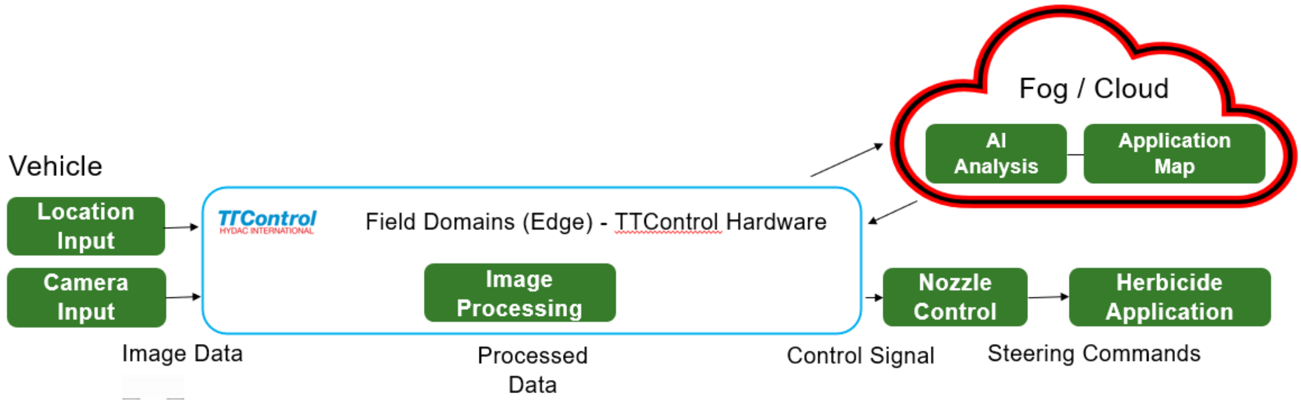


Figure 146. Cloud adoption

The Fog/Cloud diagram is shown in Figure 147 below:

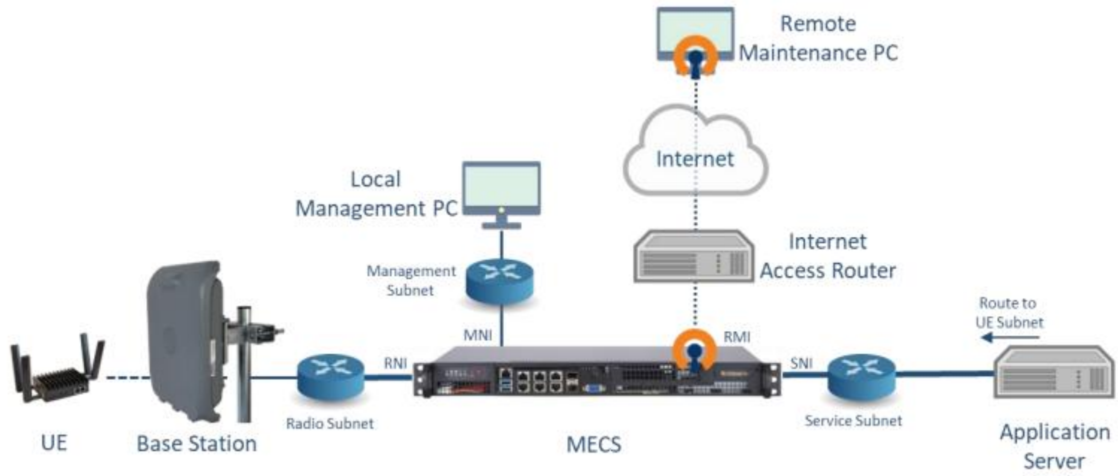


Figure 147. MECASware campusXG network with Application Server

This infrastructure, supplied by the company **MECSware**, enables the execution of **AI analysis** and the creation of the **application map** on the local application server (same server as in the Lab setup), and serves as the communication layer to the field domain with the respective User Equipment (UE) (MECS-UE-1204-MOQ3.

System Features – 5G Network – Base Station

Table 15. Technical Data 5G private Network from MECSware.

Category	MECSware campusXG®
Network Technologies	5G, 4G
3GPP RAN Option	Integrated RAN
System Deployment Model	All-on-Premise

MECS Hardware Variants	19" rack, DIN rail
Base Station Hardware Variants	Indoor, Outdoor
Network Synchronization	GPS, PTP-L2, PTP-L3 (IEEE 1588v2)
Deployability / SON	Auto-connection, ANR, Static/DHCP UE IP allocation, Dynamic Cell Configuration (DCC)
Scalability	Control and User Plane Separation (CUPS), Network slicing
Security	SIM cards, eSIM profiles, SUCI key management, SSD/HDD encryption, Secure Boot, Firewall
Operability	MECS Dashboard (Web GUI), MECS API (RESTful), Remote management (OpenVPN), User roles
Diagnostics	Throughput (iPerf3), Latency, Handover relations, Packet capture (PCAP)
User Application Hosting (MEC)	KVM virtualization infrastructure
MECS Apps	Network Performance Monitor, Log Management System, Transport Network Management System

System Features – Communication devices (UE)

Table 16. Technical data User Equipment

Parameter	Specification
Air Interface	5G Band n78
Interfaces	2× GE RJ-45, 1× USB 3.1
Antenna Connectors	VESA: 4× SMA(f); DIN rail: 2× SMA(f)
CPU	Intel 4C/4T
Memory	16 GB
Storage	256 GB SSD
Virtualization for MECS Apps	KVM
Management	MECS Dashboard (Web GUI)
Data Throughput	400 Mbps
Power Supply	9–42V DC, 15W
Dimensions (without antennas)	132.8 × 100 × 34.8 mm
Weight (without antennas)	420 g
Mounting Options	VESA, DIN top-hat rail
Operating Temperature	0°C–45°C (optional: -20°C–70°C)
Ingress Protection	IP30
Vibration Resistance	MIL-STD810, 7M1

2.3.1.2.2. Development activities

Pilot3-BusinessProcess1-DevelopmentActivity-13 (P3-BP1-DA13) Development Image Processing Tool:

The method developed for AI-based weed detection is grounded in the principles of **object detection** within image processing. This approach enables the system to identify the **precise location** of objects—such as weeds—within an image by generating **bounding boxes** and assigning each detected object to a predefined class. The output consists of a compact representation including the bounding box coordinates (x, y, width, height) and the class label, making it highly suitable for downstream control tasks such as **precision spraying**.

Figure 148. Samples from validation Datasets

Unlike image classification, which only determines whether an object class is present in an image, object detection provides **spatial localization**, which is essential for agricultural applications where targeted interventions are required. This capability allows for selective treatment of specific areas, reducing input usage and environmental impact. In this image processing tool **YOLO (You Only Look Once)** has been used.

In practical agricultural deployment, cameras mounted on tractors or drones continuously capture image data. The AI system processes these images in real time, detects and localizes weeds, and generates a **georeferenced application map** using GNSS data. This map is then used to guide autonomous spraying systems, enabling **precise, resource-efficient, and automated field operations**.

Part of the AI tool is the compression of the images to generate a leaner processing. This is done in the Edge node. By using the orchestration provided by the aeroS framework, the AI Model will be executed in the cloud domain. The generated map will then be distributed to the machines in the field.

Pilot3-BusinessProcess1-DevelopmentActivity-14 (P3-BP1-DA14) Development Spraying Adaptation Application:

The spraying adaptation application was developed to dynamically adjust spraying parameters based on environmental and crop conditions, enhancing precision and efficiency in agricultural treatments

The spraying adoption application utilizes the georeferenced application map derived from AI-based weed detection (described in P3-BP1-DA1) to enable **targeted and autonomous spraying** in agricultural fields. The process begins with high-resolution image data captured by cameras.

Each detection is tagged with precise GNSS coordinates, allowing the system to generate a **spatially accurate prescription map**. This map contains metadata such as weed density, location, and treatment zones, formatted to be compatible with tractor control systems (e.g., ISOBUS or proprietary interfaces).

The spraying application interprets this map and translates it into **machine-level instructions** for autonomous execution. Depending on the tractor type (electric or diesel), the system adjusts parameters such as:

- Spray rate
- Nozzle activation
- Path planning

- Speed control

The application ensures that herbicide is applied **only where needed**, minimizing chemical usage and environmental impact. It also supports **variable-rate spraying**, adapting the dosage based on weed intensity and spatial distribution.

Real-time telemetry and feedback loops allow the system to monitor execution accuracy, energy consumption (kWh/ha or liters/ha), and coverage efficiency.

Pilot3-BusinessProcess1-DevelopmentActivity-17 (P3-BP1-DA17) Development Tracking and Navigation Application:

The **tracking and navigation application** was developed to tag precise real-time positioning alongside the captured image data. Latitude and Longitude coordinates in WGS84 coordinate system is used to tag the video images. As a sensor the The StarFire™ 7500 Receiver is in use. It is transferred via ethernet to the field domain. The data is also available on the vehicle CAN bus for steering applications and documentation. It is used for autonomous path planning of agricultural machinery in dynamic field environments, as outlined in this pilot (Figure)

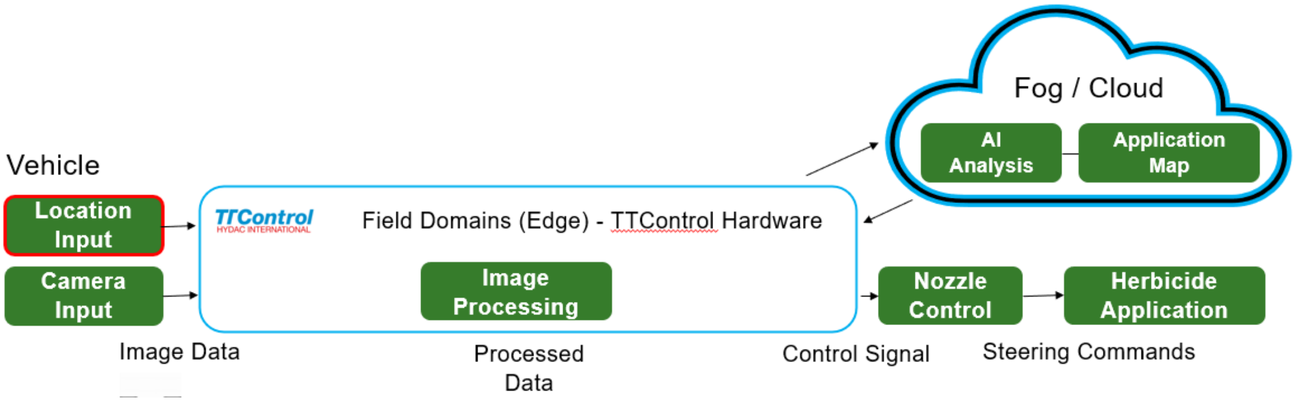


Figure 149. Location Device Setup

The sensor used is the John Deere **StarFire™ 7500 Receiver**. The StarFire™ 7500 Receiver is a high-precision GNSS receiver designed for advanced agricultural guidance and automation. It supports multiple satellite constellations and offers RTK-like accuracy without requiring additional hardware. The receiver is ideal for operations such as planting, spraying, and harvesting, and integrates seamlessly with John Deere's AutoTrac™ systems. See Table below for detailed specification.

Table 17. Specification StarFire™ 7500 Receiver

Feature	Specification
Correction Signal Options	SF1, SF-RTK, Radio RTK, Mobile RTK
Accuracy (SF-RTK)	±2.5 cm horizontal pass-to-pass
Accuracy (SF1)	±15 cm horizontal pass-to-pass
Satellite Constellations	GPS, GLONASS, Galileo, BeiDou
Pull-In Time (SF-RTK)	Up to 73% faster than SF3 (approx. 3× faster than StarFire 6000)
Repeatability	Up to 5 years season-to-season (SF-RTK)
Terrain Compensation Module (TCM)	Integrated; adjusts for roll, pitch, and yaw
Signal Switching	Automatic switching between StarFire satellites for optimal performance
Connectivity	Ethernet (for over-the-air updates), JDLink™ compatible

Security Features	Optional locking bracket for theft protection

Pilot3-BusinessProcess1-DevelopmentActivity-4 (P3-BP1-DA14): Identified issues with operating system version of the field domain (TTC Motion ECU from TTControl).

By default, the ECU comes with Linux kernel v4.9 which is incompatible with kernel patches **to enable self-realtimeness functionality**. Furthermore, IESE identified missing kernel modules possibly required for a clean deployment of Kubernetes. A software update for ECU has been provided by TTC featuring Linux kernel v5.10 that is compatible with self-realtimeness kernel patch. The self-realtimeness kernel patch requires a rebuild of the Linux kernel during which previously identified missing kernel modules to fulfill Kubernetes requirements are also compiled and added.

2.3.1.2.3. Integration activities

Pilot3-BusinessProcess1-IntegrationActivity-1 (P3-BP1-IA1): Integrate data of Sc1 in the promised flow:

The integration of data using AI for weed detection with the aerOS components is enhancing collaborative workflows in large-scale agricultural production by enabling analysis of field imagery, automating weed identification, and supporting data-driven decision-making across multiple farming operations and machines.

The video stream captured by the Ethernet camera constitutes a continuous recording of a conventional farm field located in Germany including sections of soil, grass, and weed (see Figure). Each and every image frame of the video footage is pre-processed on the ECU (**aerOS Entry/Field Domain**) to increase data transmission efficiency before it is actually transferred via its network interface to the high-performance workstation (**aerOS Cloud Domain**) for conducting weed detection and sprayer application map derivation.

During the preprocessing phase, the individual full-scale video frames are extracted and encoded in JPEG format allowing for further compression options in terms of image resolution and quality. Initial experimentation results have shown that image resolution downscaling from 3840x2160 to 1280x720 pixels in combination with a reduction of the JPEG image quality offers balancing parameter to tradeoff between weed detection accuracy and frame transmission speed.

The number of video frames aggregated within a single application-layer message burst is flexibly configurable, which allows for supporting different use case scenarios in terms of driving speed and sprayer application of the tractor/implement fleet participants.

As subsequent part of the distributed AI service operation, proprietary algorithms on the cloud node analyze each arriving frame and infer a weed location map based on GPS coordinates and an AI model (Rumex) pretrained with historical real-world weed detection data through federated learning. Afterwards, the synthesized value tuples are sent back to the ECU to apply that telemetry information for accurate application of herbicide in a further step, which, however, is beyond the scope of KPI evaluation in Pilot 3.

For application-layer data exchange between those two computing nodes, a WebSocket-based network connection is established during aerOS service deployment and instantiation (see Figure), which remains open for the entire duration of any evaluation trial based on a specific test parameter configuration.

Figure 150. Exemplary image frame of the recorded video footage showing a farm field with sections of soil, grass, and weed. Each frame serves as input for the distributed Rumex weed detection service, the output of which provides a weed location map with spraye

Pilot3-BusinessProcess1-IntegrationActivity-18 (P3-BP1-IA18) aerOS Basic Components

The introductory aerOS overview documents were used in conjunction with the official aerOS documentation and further publicly available resources to prepare the deployment of aerOS basic components on the two aerOS domains as per Figure . The instantiations of these domains are meant to be hosted and evaluated in an isolated laboratory environment as well as in-field. Therefore, the HPC provided by John Deere needed reconfiguration to comply with the allocated lab's network infrastructure and fulfill the prerequisites to properly run any aerOS components on a Kubernetes cluster in the first place. In this context, the Kubernetes configuration preinstalled on the HPC needed to be adapted, and the k8s cluster was reinitialized, which subsequently allowed to successfully deploy all aerOS basic components on the HPC, henceforth representing the aerOS Cloud Domain. To this end, any inter-component dependencies had to be considered to perform that iterative deployment on a component-by-component basis. Figure shows all the service pods deployed in the K8s-based Kubernetes cluster of the HPC-based aerOS Cloud Domain.

The same iterative procedure was then applied to prepare and deploy those aerOS basic components onto the aerOS Entry/Field Domain. Therefore, the ECU provided by TTControl needed to be re-flashed with a version of TTC firmware compatible with any aerOS components, which were anticipated to be deployed. In this context, it has been shown that installing an operational Kubernetes cluster is generally viable on a resource-constraint device like that ECU. Figure 113 shows all the service pods deployed in the KubeEdge-based Kubernetes cluster of the Laptop/ ECU-based aerOS Entry/Field Domain.

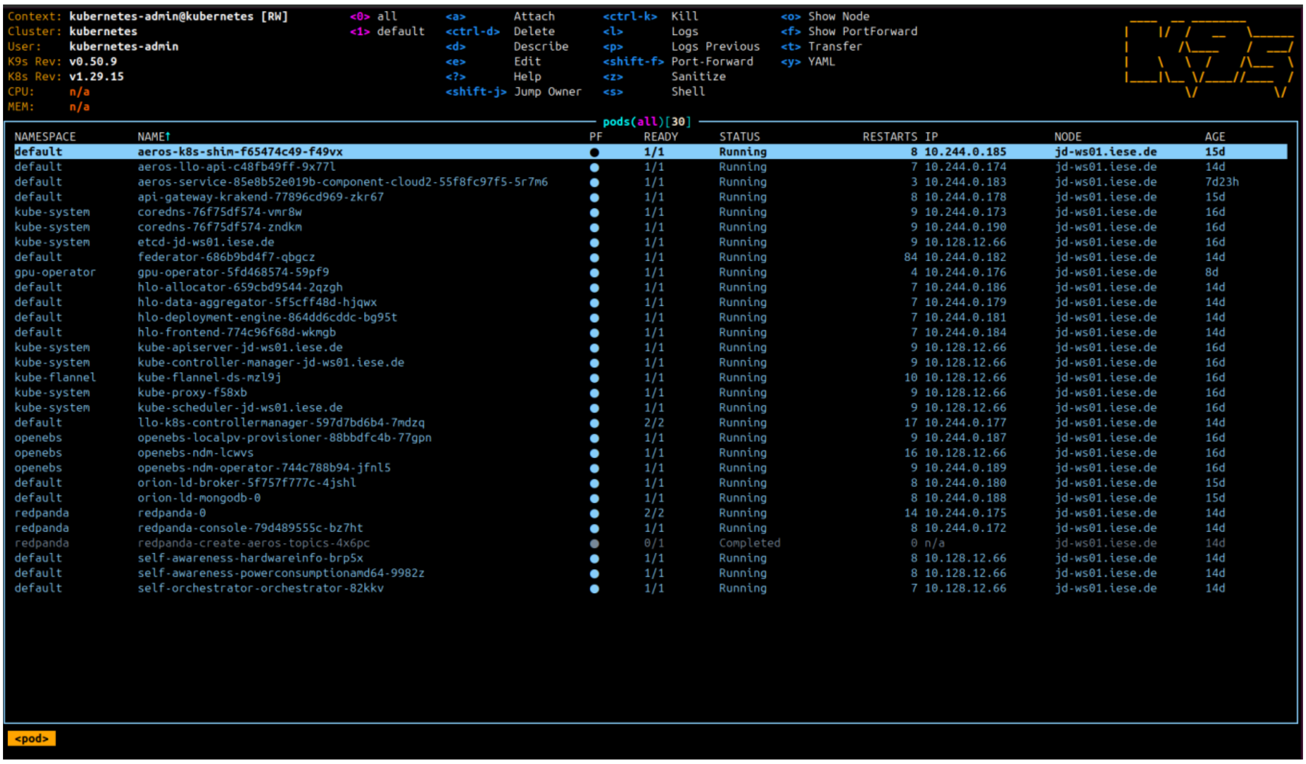


Figure 151. List and status of service pods deployed in the Kubernetes cluster of the HPC-based aerOS Cloud Domain hosting any aerOS basic and additional components relevant for Pilot 3.

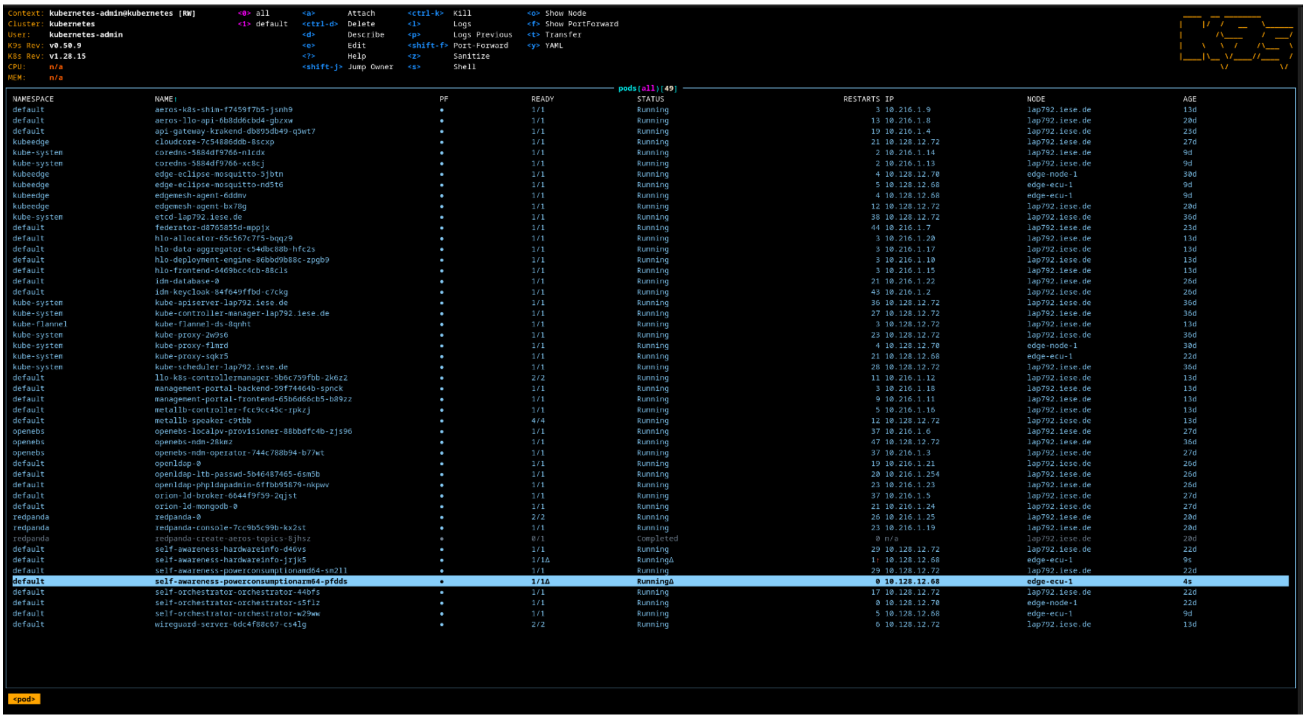


Figure 152. List and status of service pods deployed in the Kubernetes cluster of the Laptop/ECU-based aerOS Entry/Field Domain hosting any aerOS basic and additional components relevant for Pilot 3.

Pilot3-BusinessProcess1-IntegrationActivity-3 (P3-BP1-IA3) aerOS Non-Basic Components

In parallel to the preparation and deployment of aerOS basic components, any non-basic aerOS components have been deployed on the respective devices, which have been identified to be relevant in the context of the upcoming KPI evaluation. Therefore, in analogy to handling basic aerOS components, also those additional aerOS components have been prepared and deployed within the Entry/Field and Cloud Domain accordingly. The results of this activity can be derived from Figure and Figure . With respect to the AI-based weed detection service, the corresponding two components have been deployed via the aerOS Management Portal (see Figure).

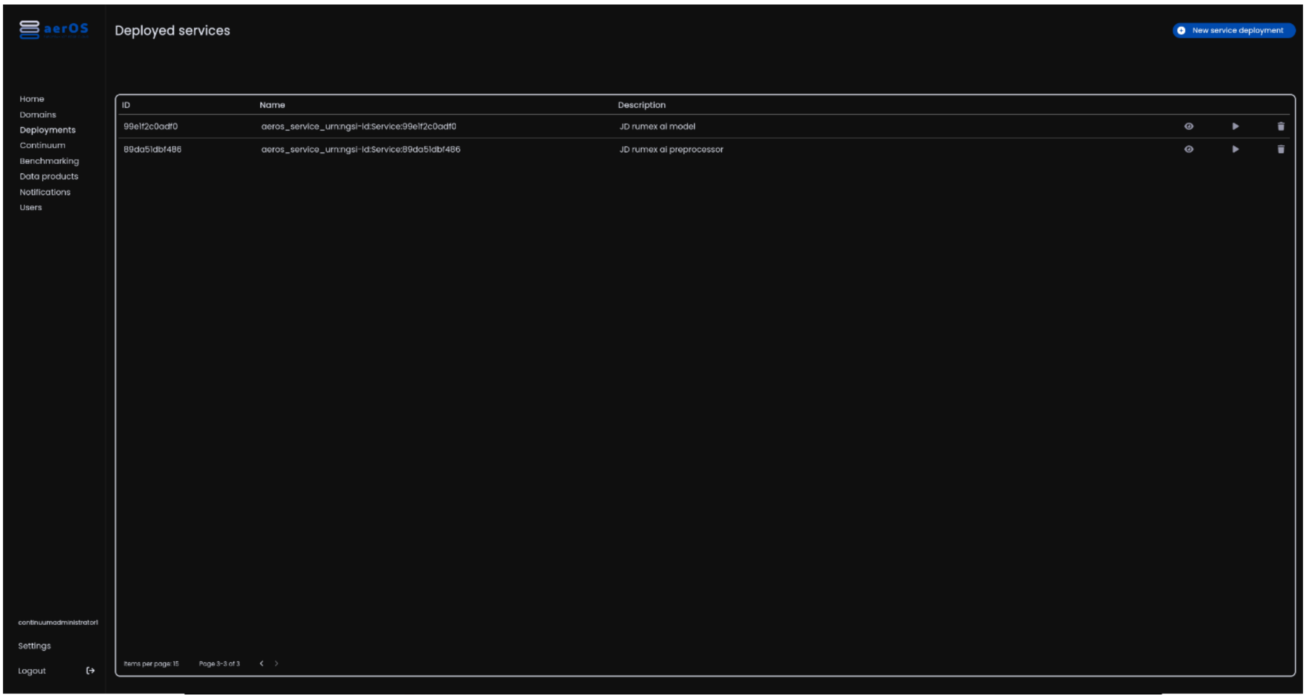


Figure 153. Graphical user interface of the aerOS Management Portal listing the JD Rumex AI Application, which is split into two distributed services, AI Model and AI Preprocessor, and deployed on the two aerOS domains, Cloud Domain and Entry/Field Domain.

Pilot3-BusinessProcess12-IntegrationActivity-20 (P3-BP12-IA20) Integrating AI Models for field operation and orchestration (using AI for weed detection):

Among the additional aerOS components, the Rumex AI model component provided by JD has been investigated in terms of intended functionality and its integrability into the overall aerOS setup in question. To this end, the required NVIDIA GPU operator was created for Kubernetes clusters and the corresponding K8s and Docker integration environment was prepared. As planned, the AI-based weed detection service was successfully deployed onto the K8s-based Kubernetes cluster of the HPC-based aerOS Cloud Domain, which can use either its dedicated GPU or its CPU for model inference execution. Since general comprehension on processing capabilities along with initial tests in the laboratory environment confirmed the assumption that CPU-based execution of the AI model service is significantly slower than GPU-based execution.

In the context of integrating the AI model for field operation, preliminary testing campaigns have been performed to investigate the impact of different input parameter settings on its performance. Interestingly, evaluation results have shown that decreasing the JPEG image quality of the 3840x2160 pixels video frames does not have any negative impact on weed detection accuracy. However, this reduction has a significantly positive effect on transmission speed. This circumstance becomes evident when comparing the exemplary but representative evaluation results regarding the average number of transmitted frames per second in conjunction with total number of detections in Figure with the same in Figure . While the detections remain constant, the transmission speed increases by a factor of 2.56.

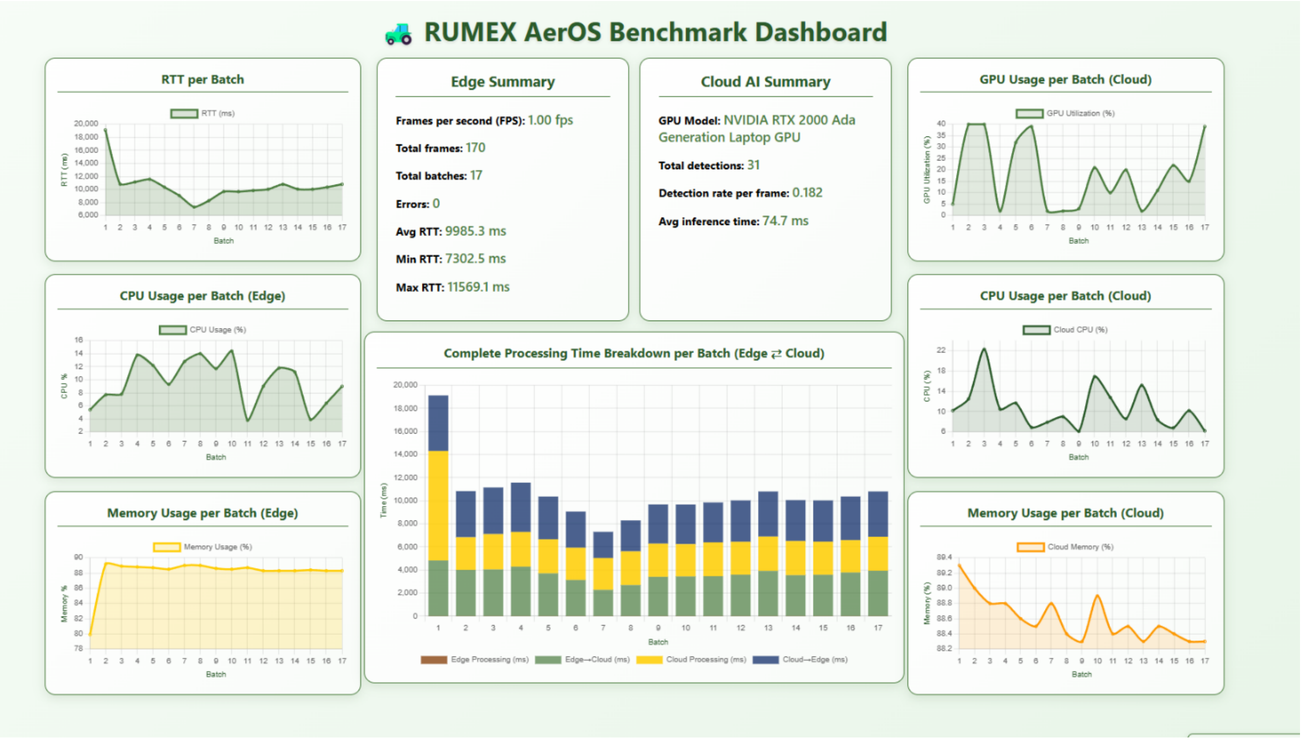


Figure 154. Evaluation dashboard visualizing benchmarking results in terms of the selected metrics for KPI substantiation in Pilot 3. The outcome shown refers to video image frames with 3840x2160 pixels resolution and JPEG image quality of 100%.

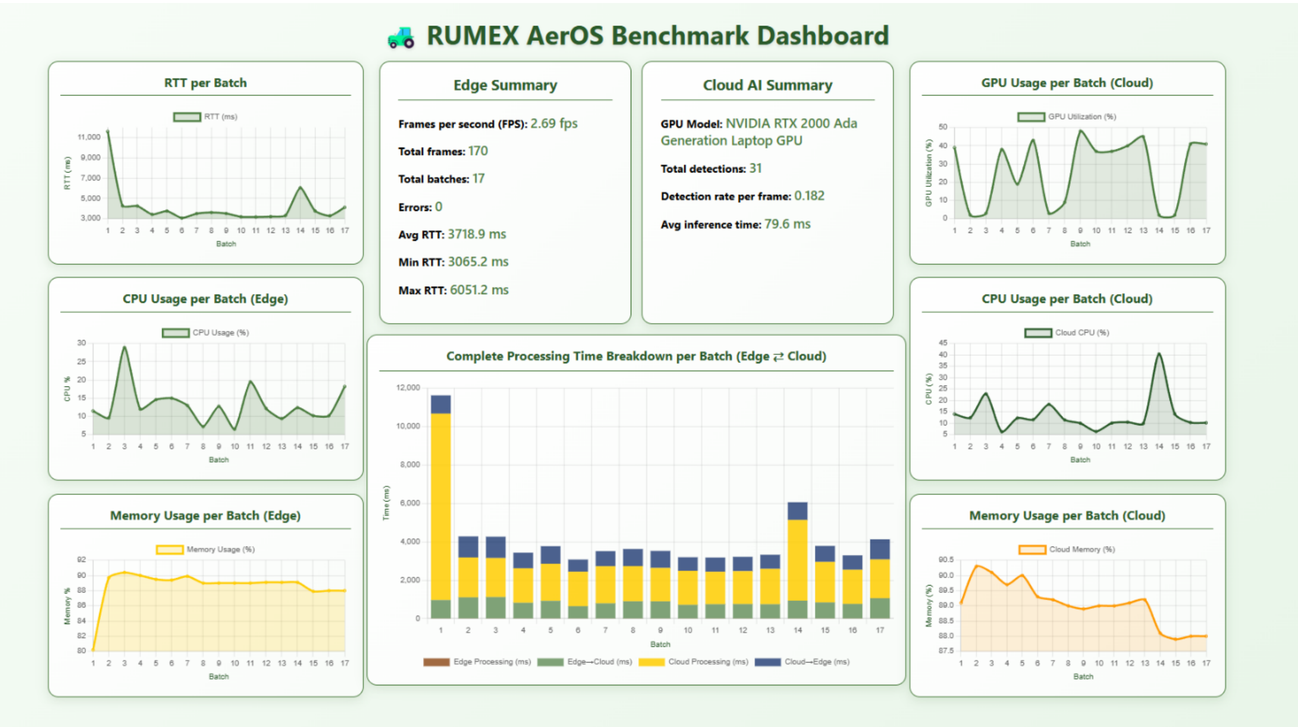


Figure 155. Evaluation dashboard visualizing benchmarking results in terms of the selected metrics for KPI substantiation in Pilot 3. The outcome shown refers to video image frames with 3840x2160 pixels resolution and JPEG image quality of 70%.

Pilot3-BusinessProcess1-IntegrationActivity-21 (P3-BP2-IA21) Simulate distribution of work data to multiple machines for operation optimization

The applied evaluation setup based on that testbed architecture is used as basis for systematic evaluation of the aerOS use case in Pilot 3 against selected key performance indicators including vehicle performance improvement (KPI 2.3.2) and emission reduction (KPI 2.3.3). The diagram in Figure details the deployment of aerOS basic and additional components including the use case-specific AI application service of John Deere, which are distributed among two aerOS domains. These domains include the Entry/Field Domain hosting a KubeEdge-based Kubernetes cluster and the Cloud Domain, which hosts a K8s-based Kubernetes cluster. Both service clusters comprise the required basic and additional aerOS components to implement a distributed service continuum featuring vertical scaling in terms of potentially multiple edge and cloud nodes as well as horizontal scaling with respect to potentially further service clusters.

As described above, the setup successfully demonstrated that the output of the AI model for weed detection—a spatially precise application map—can be distributed to multiple machines simply by adding the required connection endpoints to the distribution list of aerOS. This capability is further emphasized in the Figure below: The distribution time back to the vehicle was measured at 221 ms. With multicast functionality you can distribute the data back to the machine simultaneously.

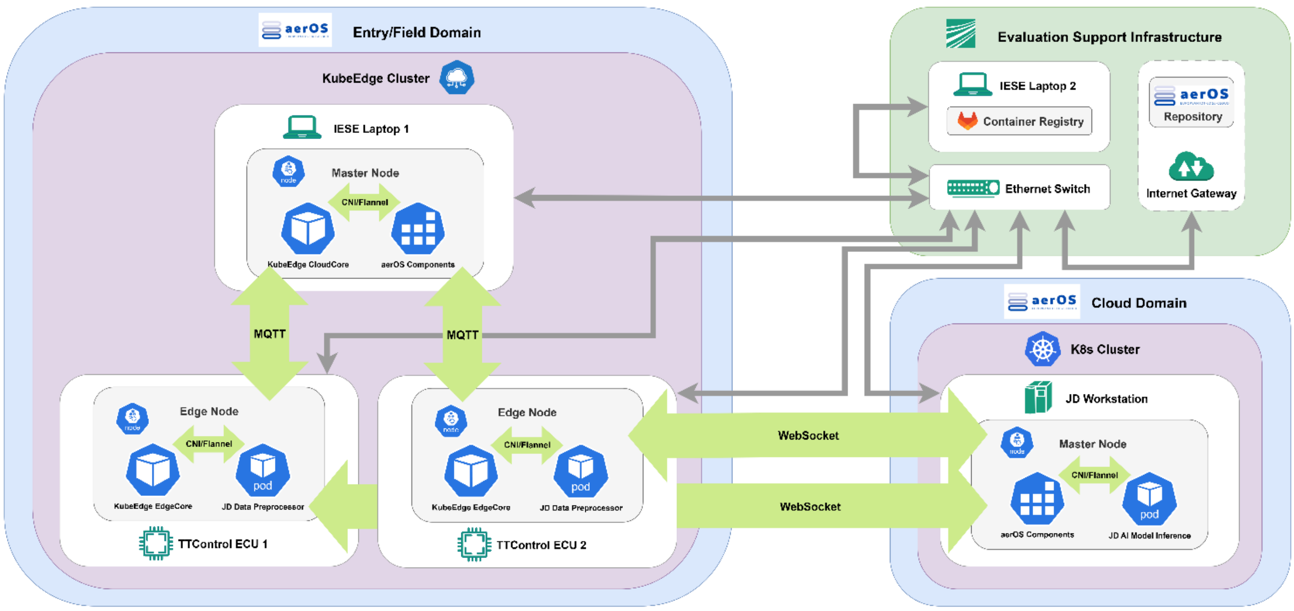


Figure 156. Overview map distribution multiple machines

2.3.2. Basis for CO2 neutral intelligent operation (Scenario 2)

2.3.2.1. Technical Schema

Same schematic as explained at the previous scenario.

Pilot 3 operates within two aerOS Domains, as illustrated in Figure 157.

Domain01 is the pilot's entry domain. It includes all aerOS services required for the administration of the pilot continuum and comprises two infrastructure elements: the Master Node and the Edge Node.

- The Master Node manages the aerOS continuum.
- The Edge Node handles video compression in preparation for the AI application in the cloud.
- Communication between the nodes is facilitated via KubeEdge MQTT.

Domain02, the cloud domain, is responsible for executing the AI application for weed detection and generating a spatially precise prescription map.

- The connection between the field and the cloud domain is established via HTTP over WireGuard VPN.

The generated Map is used for optimized path planning and automated guidance of the vehicle resulting in a more efficient energy consumption and a reduction of CO_2, as seen in the figure below.

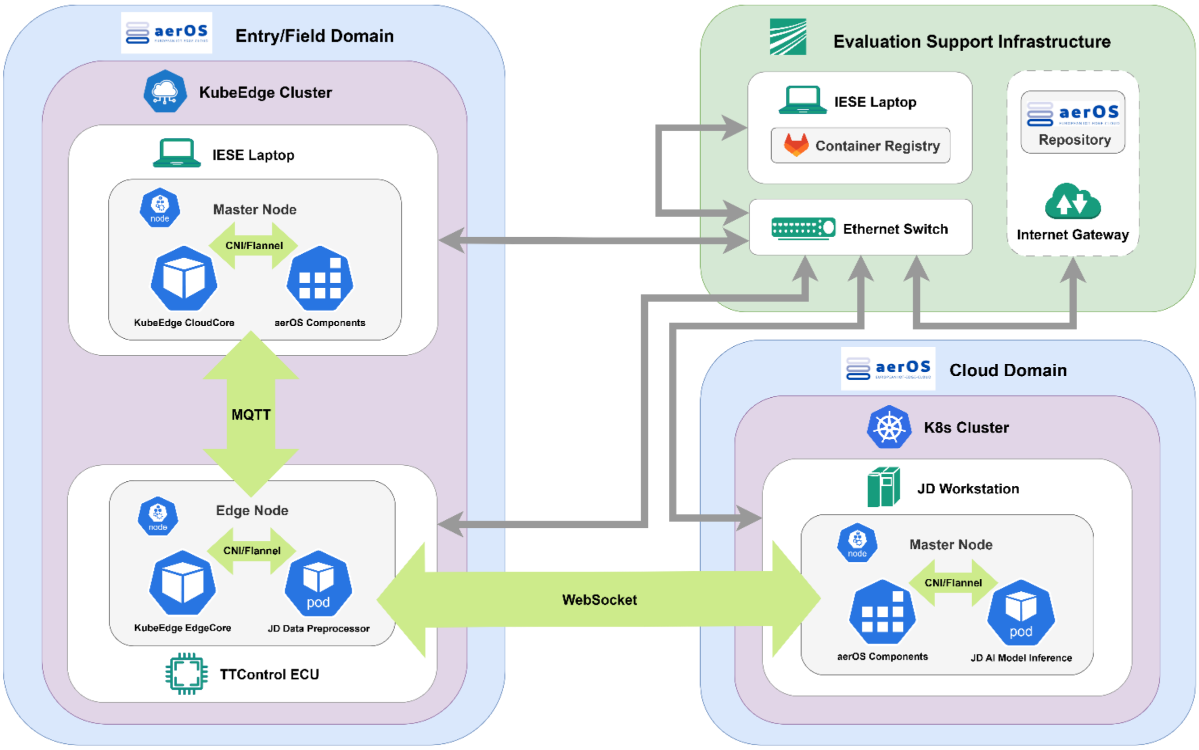


Figure 157. Deployment diagram of aerOS and supporting infrastructure components describing the evaluation testbed setup of the aerOS-based Pilot 3 use case for scenario 1 and 2.

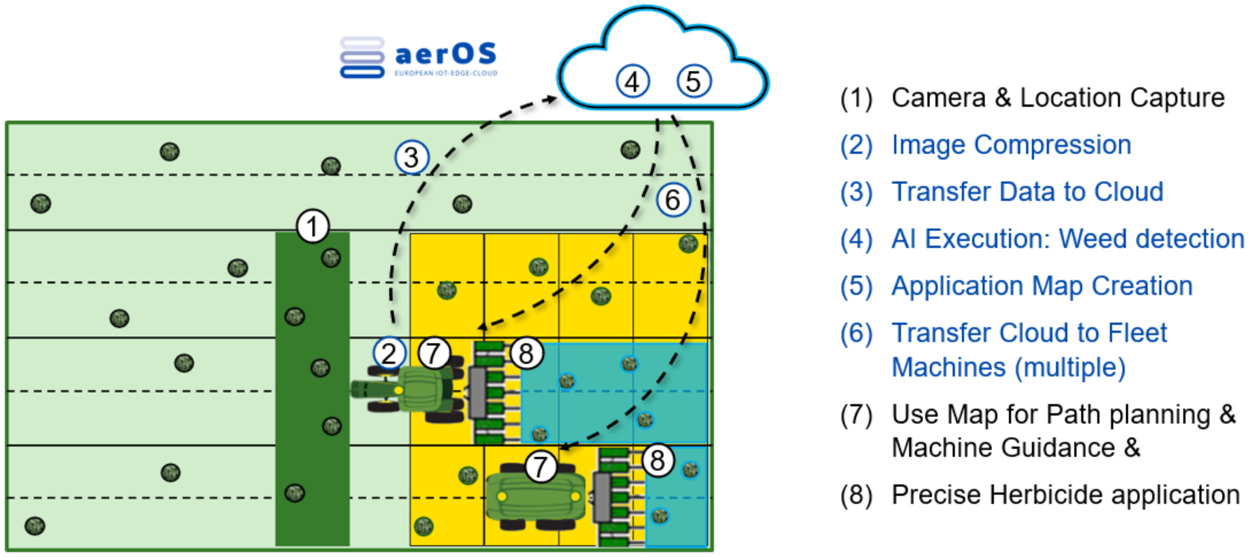


Figure 158. Cooperative Machine coordination for CO2 reduced operation

2.3.2.1. Report of activities

2.3.2.1.1. Setup and procurement activities

The procurement and setup activities listed below apply for scenario 1 and scenario 2 in the same way:

Table 18. Overview Setup and procurement activities applied for scenario 2 (Basis for CO2 neutral intelligent operations

Code	Topic	Activity
P3-BP1-SA1	Procurement ECU platform prototype	Setup
P3-BP1-SA2	Setup SESAM 2	Setup
P3-BP1-SA3	Setup Sprayers	Setup
P3-BP1-SA4	Setup StarFire GNSS Receiver	Setup
P3-BP1-SA6	Procurement Ethernet Camera	Setup
P3-BP1-SA7	Setup Ethernet Camera	Setup
P3-BP1-SA8	Procurement Computing Node	Setup
P3-BP1-SA9	Setup Computing Node	Setup
P3-BP2-SA10	Setup Extended SESAM 2 with consumption analytics (iterative with A2.1.3)	Setup
P3-BP2-SA11	Setup Adapted Computing Node (iterative with A2.1.11)	Setup

2.3.2.1.2. Development activities

Pilot3-BusinessProcess2-DevelopmentActivity-15 (P3-BP2-DA15) Vehicle Configuration Adaptation Tool:

The vehicle configuration tool serves as a critical interface for deploying and orchestrating AI models within the aerOS ecosystem, specifically tailored for agricultural field operations such as weed detection and precision spraying.

As part of Pilot 3, this tool facilitates the integration of AI components into heterogeneous computing environments across different aerOS domains (Field, On-Premise/Entry, and Cloud).

The configuration tool enables:

- **Hardware-aware deployment**: It detects available computational resources (CPU/GPU) on each domain and selects the appropriate AI model variant (GPU-accelerated or CPU-only).
- **Container orchestration**: It manages the deployment of AI models using Kubernetes and Docker, ensuring compatibility with the aerOS infrastructure.
- **Execution location flexibility**: Based on performance benchmarks and resource availability, the tool allows dynamic assignment of the AI model execution to the most suitable domain (e.g., Cloud with GPU for productive use, Field or On-Premise with CPU for demonstration).
- **Integration with NVIDIA GPU Operator**: For GPU-enabled domains, the tool ensures proper driver and runtime setup for accelerated inference.
- **KPI-driven orchestration**: The tool supports benchmarking and KPI evaluation by enabling variable execution locations, allowing comparative analysis of CPU vs. GPU performance.

This configuration tool is essential for adapting AI execution to the constraints of agricultural machinery, which may lack dedicated graphics hardware. It ensures that the AI model for weed detection can be deployed in a scalable, modular, and performance-aware manner across the aerOS continuum.

Pilot3-BusinessProcess2-DevelopmentActivity-16 (P3-BP2-DA16) Machine Analysis AI Engine Application:

The Machine Analysis AI Engine Application is designed to integrate AI-based field operation capabilities—specifically weed detection—into the aerOS computing continuum. As part of Pilot 3, this application enables intelligent orchestration and execution of AI models across heterogeneous hardware environments.

Core Functionality

The application leverages an AI model provided by John Deere, which performs object detection on image data to identify and classify weeds. This model is containerized and deployed within a Kubernetes-managed infrastructure, ensuring modularity and scalability across different aerOS domains.

Deployment Architecture

- **Cloud Domain (HPC)**:
  The AI model is deployed with GPU acceleration using a custom-developed **NVIDIA GPU Operator**. This setup allows high-performance inference for productive use cases.
- **Field Domain (ECU) & Entry Domain (Laptop)**:
  These domains lack sufficient hardware resources like a dedicated GPU. Therefore, a **CPU-only variant** of the AI model is deployed for demonstration and evaluation purposes. While functional, CPU-based execution is significantly slower and not recommended for real-time operations.

Execution Flexibility

To support performance benchmarking and KPI evaluation, the application includes logic to **dynamically assign the execution location** of the AI model. This flexibility allows comparative analysis between GPU and CPU performance and supports adaptive orchestration based on available resources.

Integration Components

- Kubernetes cluster with GPU/CPU node awareness
- Docker-based containerization of the AI model
- NVIDIA GPU Operator for driver/runtime management
- Telemetry and orchestration logic for execution control

2.3.2.1.3. Integration activities

Pilot3-BusinessProcess12-IntegrationActivity-20 (P3-BP2-IA20) Integrating AI Models for field operation and orchestration (using AI for weed detection):

As described in (P3-BP1-DA-13), an AI-based weed detection method uses **object detection** to identify and localize weeds in images via bounding boxes and class labels, enabling precise control tasks like targeted spraying. Unlike simple image classification, this approach provides spatial localization, which is crucial for selective agricultural interventions.

The system employs **YOLO (You Only Look Once)** for real-time image processing, with cameras mounted on tractors or drones capturing data. The AI processes these images, detects weeds, and generates **georeferenced application maps** using GNSS data. These maps guide autonomous spraying systems for efficient field operations.

To optimize performance, image compression is handled at the **Edge node**, while the AI model runs in the **cloud** via the **aeroS framework**. The resulting maps are then distributed to field machines for execution.

Beyond distributing the generated application map to multiple machines for swarm coordination, the map is also utilized for optimized path planning and the creation of guidance lines. This enables automated vehicle navigation and supports the generation of accurate herbicide application plans.

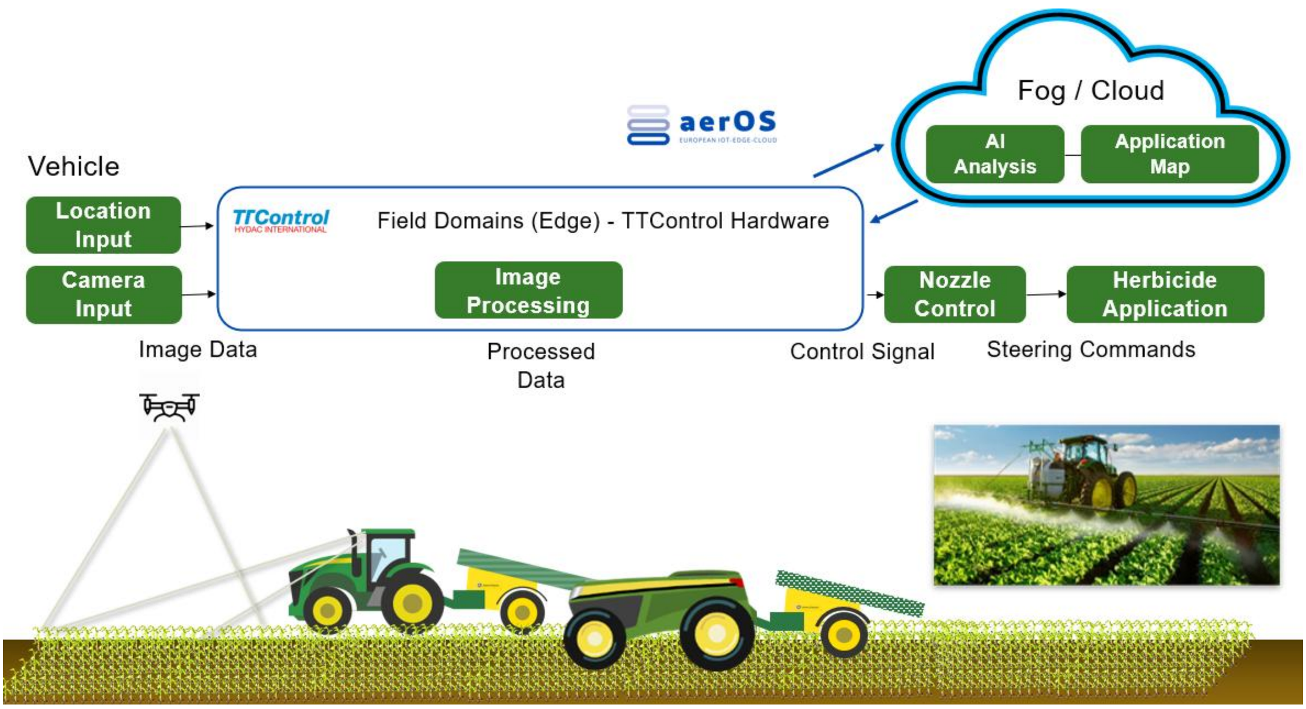


Figure 159. Field Setup overview

Pilot3-BusinessProcess2-IntegrationActivity-22 (P3-BP2-IA22) Simulate data orchestration for CO2 reduction (by integrating low latency networks):

Based on the implementation of the earOS architecture and the distribution of the AI-based weed detection model, it was possible to simulate data orchestration aimed at CO_2 reduction. The following metrics were derived from this simulation for both diesel and electric tractors.

Swarm coordination using **AI and electric/diesel tractors** can significantly reduce CO_2 emissions per hectare by optimizing:

- Spraying routes
- Overlap reduction

- Idle time minimization
- Task distribution among multiple machines

Assuming a **baseline CO_2 emission of 41.3 kg/ha** for diesel tractor spraying:

With GPS guidance (**20% reduction**): 41.3×0.80=33.0 kg CO_2/ha | 41.3-33.0 kg CO_2/ha → **Savings: ~8.3 kg CO_2/ha**

With basic overlap reduction (2–7%): 41.3×0.93=38.4 kg CO_2/ha | 41.3-38,4 kg CO_2/ha → **Savings: ~2.9 kg CO_2/ha**

With swarm coordinated ground spraying (15%): 41.3×0.85=35.1 kg CO_2/ha | 41.3-35,1 kg CO_2/ha → **Savings: ~6.2 kg CO_2/ha + Reduced Chemical Usage (40% less)**

More information about the validation activities of this pilot after the field trials will be provided in the forthcoming deliverable D5.6. Evidences of such experiment will also be included in the document.

2.3.3. Pilot 3 Time-plan

The following Gantt diagram list all the activities from M19 (February 2024) until M35 (July 2025), representing evolution from status declared in D5.3.

Table 19. Pilot 3 updated Gantt timeline

PILOT 3		2024											2025								
Code	**Name**	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33	M34	M35	M36	M37	M38
Business Process 1 (Scenario 1) - Cooperation large Scale Production																					
Setup & Procurement Activities																					
P3-BP1-SA2	Setup Ethernet-based ECU platform prototype																				
P3-BP1-SA3	Setup SESAM 2																				
P3-BP1-SA4	Setup self-propelled sprayer																				
P3-BP1-SA8	Setup Ethernet Camera																				
P3-BP1-SA9	Procurement Computing Node (on premise domain)																				
P3-BP1-SA10	Setup Computing Node (on premise domain)																				
P3-BP1-SA11	Setup Extended SESAM 2 with consumption analytics (iterative with A2.1.3)																				
P3-BP1-SA12	Setup Adapted Computing Node (iterative with A2.1.11)																				
Development Activities																					
P3-BP1-DA1	Development Image Processing Tool																				
P3-BP1-DA2	Development Spraying Adaptation Application																				
P3-BP1-DA3	Development Tracking and Navigation Application																				
P3-BP1-DA4	Ethernet-based ECU Platform Prototype setup and integration																				
Integration Activities																					
P3-BP1-IA2	aerOS Basic components																				
P3-BP1-IA3	aerOS Non Basic components																				
P3-BP1-IA4	Integrating AI Models for field operation and orchestration (using AI for weed detection)																				

Validation Activities	
P3-BP1-VA1	*KPI Validation (Lab)*
P3-BP1-VA2	*KPI Validation (Real)*
P3-BP1-VA3	*KPI TTC Validation*
Business Process 2 (Scenario 2) - Basis for CO2 neutral intelligent operations	
Setup & Procurement Activities	
P3-BP2-SA1	Setup Extended SESAM 2 with consumption analytics (iterative with A2.1.3)
P3-BP2-SA2	Setup Adapted Computing Node (iterative with A2.1.11)
Development Actvities	
P3-BP2-DA1	Vehicle Configuration Adaptation Tool
P3-BP2-DA2	Machine Analysis AI Engine Ap-plication
Integration Actvities	
P3-BP2-IA1	Integrating AI Models for field operation and orchestration (using AI for weed detection)
P3-BP2-IA2	Simulate data orchestration for CO2 reduction (by integrating low latency networks)
Validation Activities	
P3-BP2-VA1	*KPI Validation (Lab)*
P3-BP2-VA2	*KPI Validation (Real)*

2.4. Pilot 4 - Smart edge services for the Port Continuum

Pilot 4 demonstrates the integration of aerOS in a real-world port logistics environment at the Port of Limassol, focusing on predictive maintenance and risk prevention via edge-based computer vision. Led by PRODEVELOP, EGCTL, and CUT, the pilot addresses safety and operational efficiency in container handling. aerOS components such as *Self- modules*, HLO, and Federator are deployed to enable real-time monitoring, secure orchestration, and autonomous decision-making. The pilot validates aerOS's capability to support edge intelligence in safety-critical, high-throughput industrial settings.

Pilot 4 has been planned in three phases, with their corresponding milestones. A Gantt chart complementing the above information is included in section 2.4.3

- **Pilot 4 - Milestone 1 (P4M1):** The first milestone was related to the procurement and acquisition of all the required HW and SW licenses for project's experimentation. Although originally was planned to have all of them available before mid-term of the project, unforeseen shipping delays required 6 months for most of the components, i.e., August 2024. In addition, due to the lack of availability, the latest IEs were acquired and installed during the first quarter of 2025.
- **Pilot 4 - Milestone 2 (P4M2):** The second milestone was related to the first tests with all the custom software components developed for the pilot, as well as initial integration of the aerOS core services. While the majority of the basic aerOS services were deployed by M36, i.e., August 2025, not all of them, as well as pilot-specific ML models were ready.
- **Pilot 4 - Milestone 3 (P4M3):** Finally, a third phase, including further development refinements, all aerOS services deployment, integration, and verification, is being reached by M38, i.e., October 2025.

The following table showcases how Pilot 4 has switched from processing data in the Cloud to the Edge.

Table 20. Pilot 4 Edge vs Cloud comparison before and after aerOS.

	Cloud	Edge
Before aerOS	Only one central server with CMMS and TOS was available at EUROGATE premises, without any control and orchestration of the different assets at the terminal.	None of the assets of the terminal were digitalized when the project started, so there was no possibility of condition monitoring in 2021.
After aerOS	A cloud instance, plus two additional servers in EUROGATE and CUT have been deployed, allowing for a suitable digitalization of the port, with novel added-value AI services such as predictive maintenance and Computer Vision, as well as full integration with the CMMS. It has been decided not to integrate TOS due to security reasons. However, it is being discussed when and how it can also be integrated in the platform in order to add new scenarios to potentially improve operational efficiency, environmental and safety conditions at the terminal.	There are in total 6 far-edge IEs in the port: (i) the two STS cranes and two straddle carriers whose data is being collected in the edge and use for predictive-based edge-AI processing; and (ii) two jetson Orin nodes, which connect directly to the cameras located at the gantry of the cranes and perform without latency constraints multiple object detection (container, container ID, different type of damages, and seals). It is expected to integrate the rest of straddle carriers of the port in a near future, and additional assets are also under discussion among project partners.

2.4.1. Predictive maintenance of Container Handling Equipment

2.4.1.1. Technical schema

Pilot 4 has been divided into 2 scenarios. The first scenario, **Predictive maintenance of Container Handling Equipment** has implemented and validated different ML algorithms in order to detect anomalies at CHEs from various data feeds (sensors, PLCs, and CMMS). The second scenario, **Risk prevention via Computer Vision in the edge** goal, was to develop and validate different Computer Vision AI-based algorithms on the edge, allowing EUROGATE to automatically identify containers with damage, and to check for the existence of container seals without the need of human intervention.

The diagram below illustrates the technical scheme of the demonstrator for the two Pilot 4 scenarios. As can be seen, the pilot conveyed 3 different aerOS domains. For Pilot 4 Scenario 1, EUROGATE domain is formed by 4 far-edge IEs located at cranes (2) and straddle carriers (2) that collect data from sensors and PLCs and perform local ML inferencing for the Scenario 1; as well as other 2 far-edge IEs that connect to the IPTV streaming and perform local ML inferencing for the Scenario 2. The other two Pilot 4 domains include the entrypoint domain deployed at a public cloud instance, which is in charge of the distribution management and orchestration services from aerOS to the public audience, and the CUT domain, which is used for more intense ML training and further testing of the CV models from Scenario 2.

To sum up, Pilot 4 edge-cloud infrastructure is balanced as follows: 6/9, i.e., 66% edge IEs, and 3/9, i.e., 33% cloud IEs.

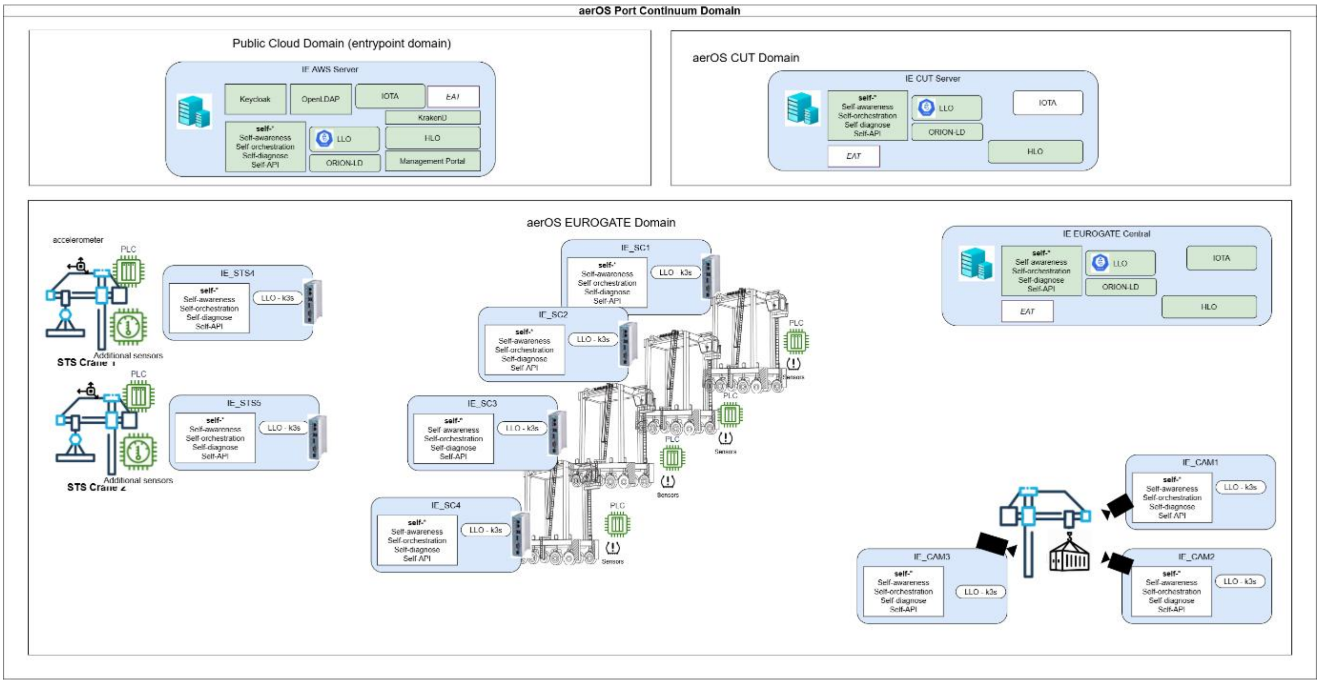


Figure 160. Pilot 4 Topology diagram – Technical Schemag

2.4.1.1. Report of activities

2.4.1.1.1. Setup and procurement activities

<u>Pilot4 – Business Process 1 – Activity - 1 (P4-BP1-SA1) STS Cranes PLCs</u>:

STS crane from OMG Bedeschi series M08140 is handled by a set of PLCs from Siemens model Simatic S7-1500. The Simatic S7-1500 is the basis for Industry 4.0, and IoT Automation by making use of the OPC-UA standard interface. They are equipped with 1516F-3 PN/DP CPU, with work memory 1.5 MB for program and 5 MB for data. It also supports PROFINET IRT protocols, with 2-port switch. ETHERNET and PROFIBUS are also interface supported. It runs jobs with speed up to 10 ns bit performance.

Figure 161. STL Crane PLC

Straddle Carriers from Kalmar series ESC 440W are equipped with PLC from Omron model CS1G-CPU44H, which are mounted with CPU Unit, with up to 40 Modules able to run 64K Words. They are provided with RS232C/USB interfaces for custom development.

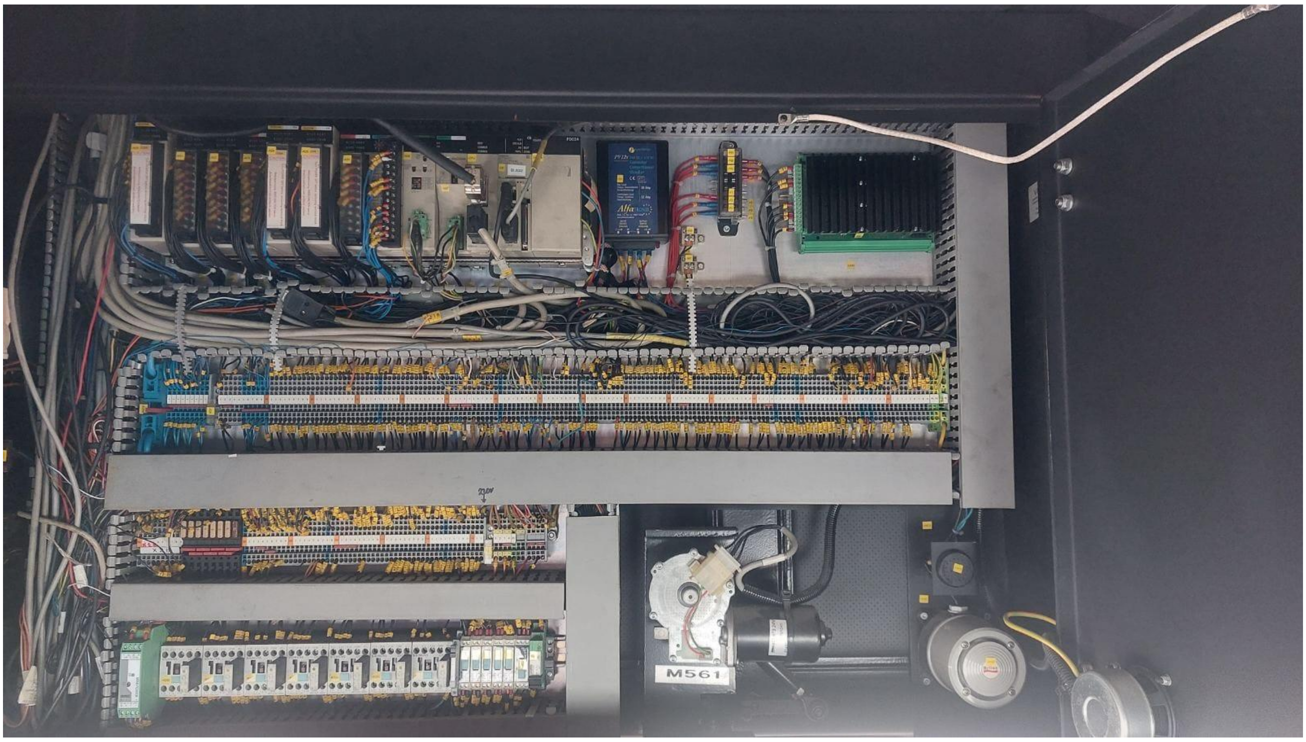


Figure 162. Straddle Carrier PLC

Pilot4 – Business Process 1 – Activity - 3 (P4-BP1-SA3) Straddle Carriers sensors:

Three main types of sensors are used in the project. In particular

- *Danforss 060G1109 hydraulic Pressure Sensors.* These sensors cover different output signals and include versions with absolute and relative pressure reference and measurement ranges between 0-1 and 0-600 bars.
- *IMUF99PL-SC3600-0KB20V1501 inclination and acceleration sensors.* The IMUF99 is optimized to provide stabilized inclination and acceleration data as well as rotation rate data. The horizontal in-clination can be reliably determined using the 3-measuring axis. The dynamic angle accuracy can be individually configured by selecting a compensation range to counteract the influence of external accelerations. Different output types are selectable for the angle definition (Euler angle, Euler vec-tor, quaternions). In addition, accelerations and rotation rates are reliably measured in the 3-measuring

axis. For further optimization of the measured value quality, filters can be set to sup-press external vibrations.

- *VIM32PL-E1V16-0RE-I420V14 vibration sensors.* It determines the vibration quantity using rms (root-mean-square) averaging. This form of quadratic averaging or pre-filtering enables precise trend statements about the condition of the application.

Figure 163. Pilot 4 Different sensors installed in the Straddle carriers under test in the project.

Pilot4 – Business Process 1 – Activity - 4 (P4-BP1-SA4) Straddle Carriers GPS:

Different advanced analytics features associated to the Straddle carrier's telemetry require the knowledge of their position. Commercial GPS has been installed, which is appended to the timely synchronized telemetry data collected from the PLC.

Figure 164. Pilot 4 GPS Antenna.

Pilot4 – Business Process 1 – Activity - 5 (P4-BP1-SA5) Straddle Carriers Human Machine Interfaces:

The Kinco HMI GT070HE is connected to the Straddle carrier PLCs, and has all the functionalities needed to access PLC. Features: 7" HD display, 16.77M true color; support Ethernet function; powerful script function, support edge calculation; built-in Iot function, support remote service.

Figure 165. Pilot 4 HMI set-up.

Pilot4 – Business Process 1 – Activity - 6 (P4-BP1-SA6) Straddle Carriers 4G Routers:

The Teltonika RUT240 is an industrial M2M/IoT 4G router that supports two Ethernet ports and a wireless interface with Hotspot functionality. It permits to publish the straddle carrier data, i.e., the internal data from the Straddle Carrier domain, to the EUROGATE and/or CUT domains. Features: 4G/LTE (Cat 4), 3G, 2G; Atheros, MIPS 24Kc, 400 MHz; 64 MB RAM; TCP, UDP, IPv4, IPv6, ICMP, NTP, DNS, HTTP, HTTPS, FTP, SMTP, SSLv3, TLS 1.3, ARP, PPP, PPPoE. It runs over RutOS which is based on Linux.

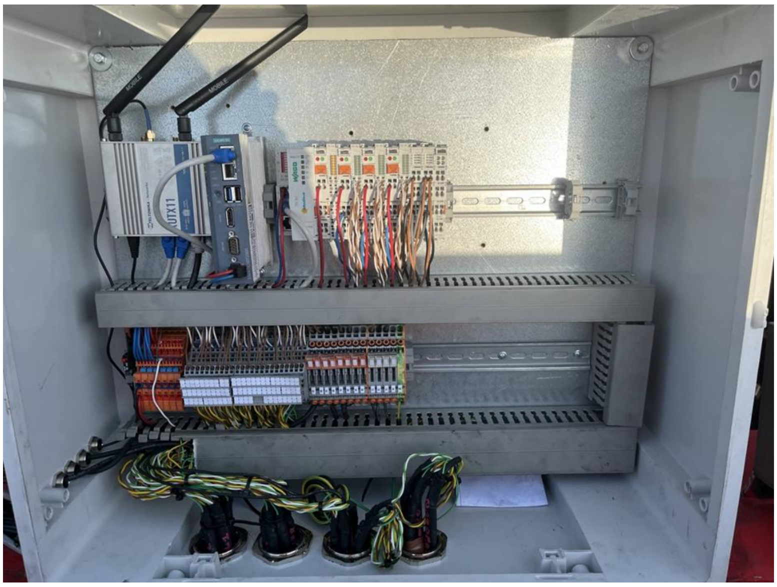


Figure 166. RUTX11 4G Router and IoT Siemens 2050 gateway installed in Straddle Carrier PLC room (see description below).

Pilot4 – Business Process 1 – Activity - 7 (P4-BP1-SA7) Straddle Carriers and STS IoT Gateways (IE1-IE4):

The Siemens Simatic IoT2050 were selected as the IoT Gateway for the pilot, due to its special focus on the industrial IoT environment and round off the SIMATIC IPC product range. It is in charge of reading, extracting, pre-processing and publishing data from the STS PLCs, Straddle Carrier PLCs, Straddle Carrier Sensors, and GPS. All these steps of data acquisition are programmed into its embedded NodeRed open-source software service on Linux. The data once processed by the IoT2050 is published through an external 4G modem explained previously. Features: Intel 1.1GHz CPU, RAM 2GB, 16GB eMMC, 2xRJ45 Ports, 1xSerial Port.

Pilot4 – Business Process 1 – Activity - 8 (P4-BP1-SA8) EUROGATE Domain Server (IE5):

This server acts as the entrance IE of the EUROGATE Domain. It provides a VM that collects all the PLCs, sensors, and GPS data acquired from STS cranes and Straddle Carriers. In addition, it also contains the orchestration and networking requirements for aerOS communication, enabling the distribution of workloads among the other 4 IEs of the domain (the IoT gateways installed on the cranes and straddle carriers under monitoring).

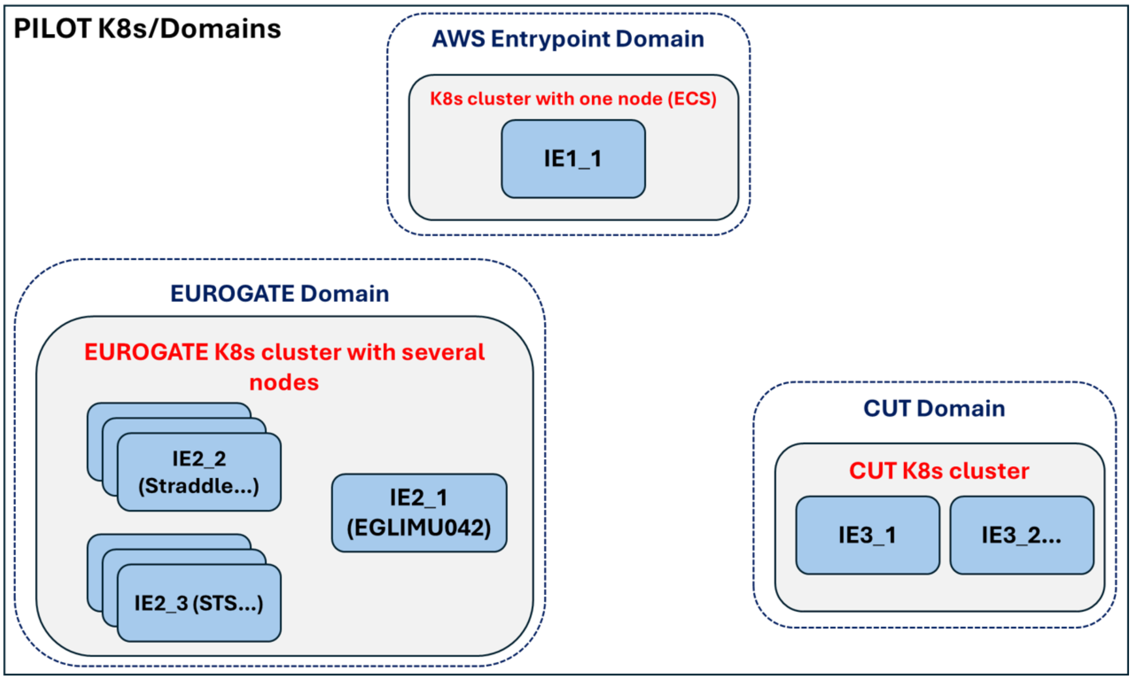


Figure 167. High-level architecture of the Pilot 4. EUROGATE Server represents IE2_1 who handles all the cranes and straddle carriers of the pilot connected to aerOS.

Pilot4 – Business Process 1 – Activity - 9 (P4-BP1-SA9) Entrypoint domain Server (IE0):

This is the entrypoint domain of the Port continuum, which is a specific instance deployed at the public cloud of Amazon Web Services.

Originally it was planned that the entrypoint domain was going to be deployed in EUROGATE premises, but due to cybersecurity concerns, it was moved to AWS.

In particular the server is an EC2 instance (4 vCPU, 16 GB RAM), with a 300 GH t3a.xlarge disk, running Ubuntu server OS. It is connected to EUROGATE and CUT domains in a site-2-site Virtual Private Connection.

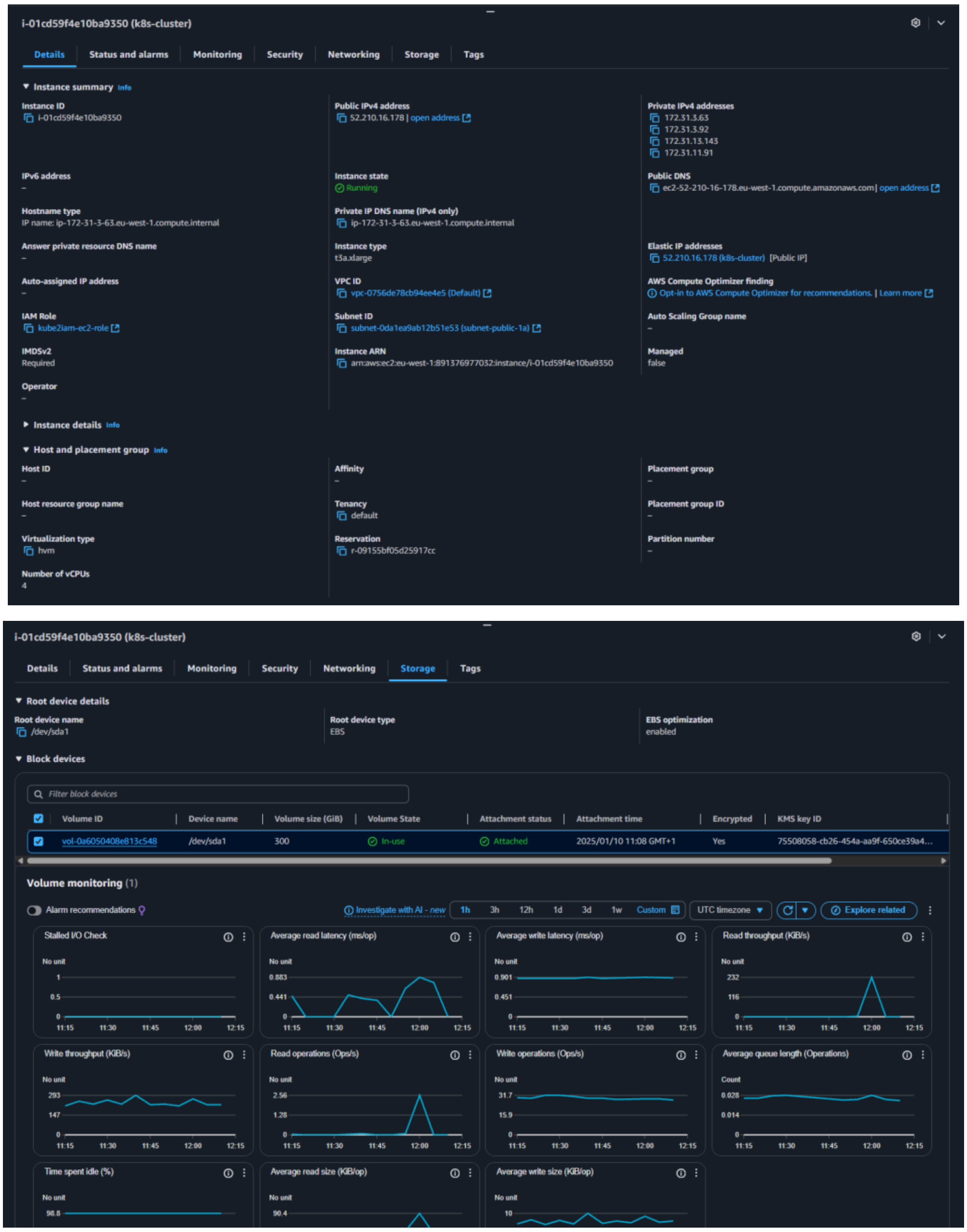


Figure 168. AWS server that integrates the Pilot 4 entrypoint domain IE.

2.4.1.1.2. Development activities

Pilot4 - Business Process 1 - Activity - 10 (P4-BP1-DA1) PLC data gathering:

This activity can be split into two-fold sub activities. On the one hand, STS crane PLC data is retrieved by the use of specific Siemens libraries running under node-red environment within the IoT Gateways. On the other hand, the straddle carrier PLC data is acquired by the HMI, which sends the data to the an EMQx MQTT broker, which is in charge of managing the different MQTT topics and allowing new EUROGATE and CUT servers' MQTT clients to subscribe to them.

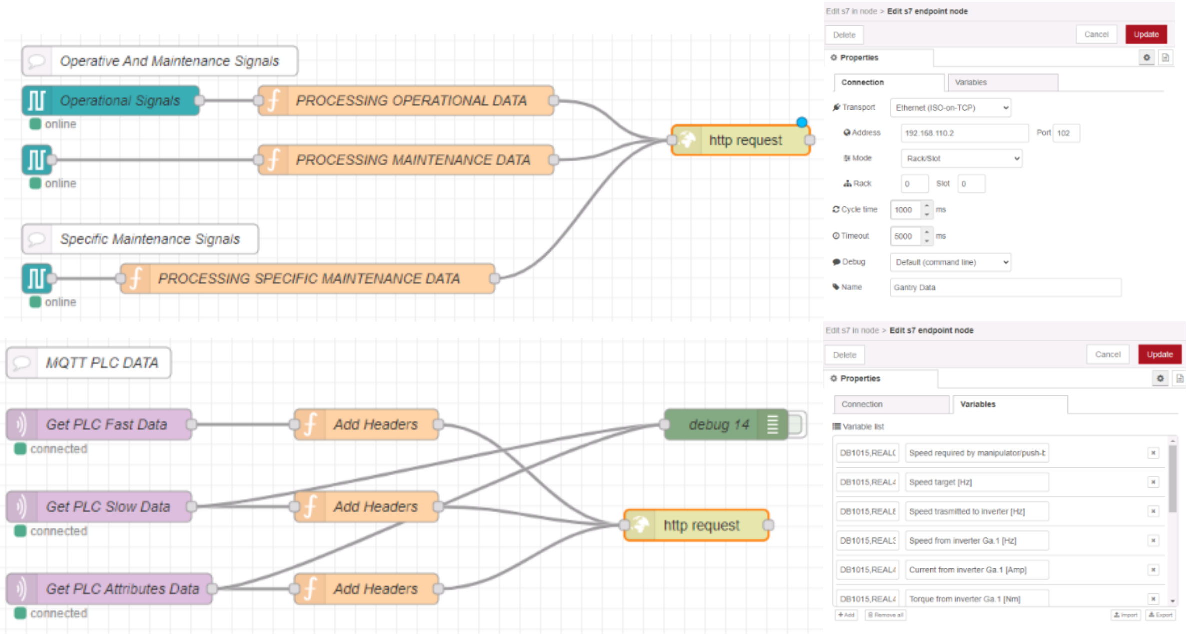


Figure 169. STP PLC Data acquisition (top) and Straddle carrier PCL Data acquisition (bottom).

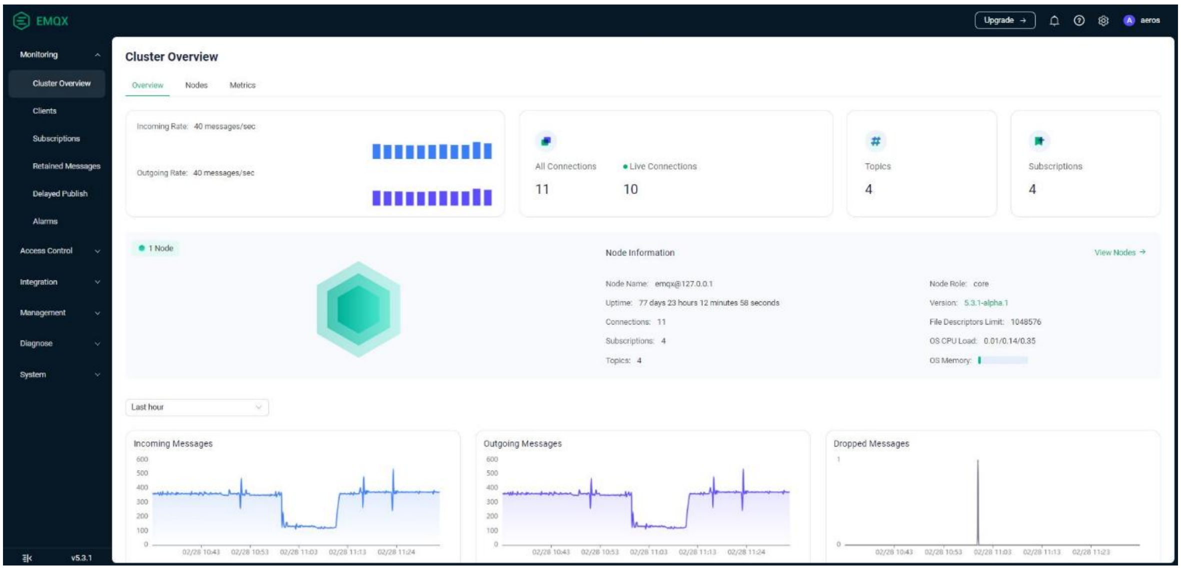


Figure 170. Pilot 4 - Straddle carrier data collection in MQTT broker.

Pilot4 - Business Process 1 - Activity - 11 (P4-BP1-DA2) GPS and sensors data gathering:

A custom component has been developed and deployed at the edge IEs of the pilot (IoT gateways). Likewise, D4.1 activities, this component is in charge of retrieving the data generated by the sensors and GPS mounted on the straddle carriers, and is transmitted to the big data platform through the MQTT broker, as shown below.

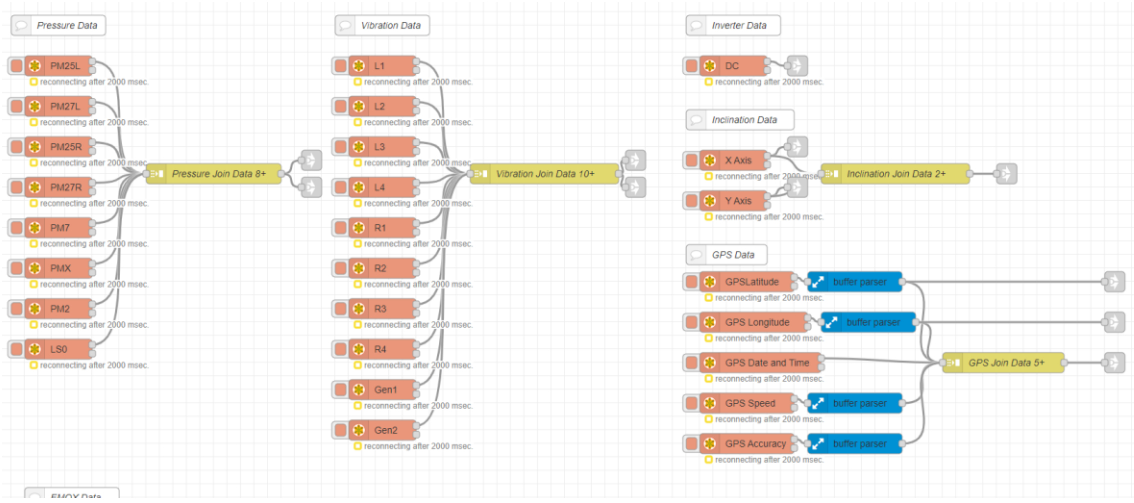


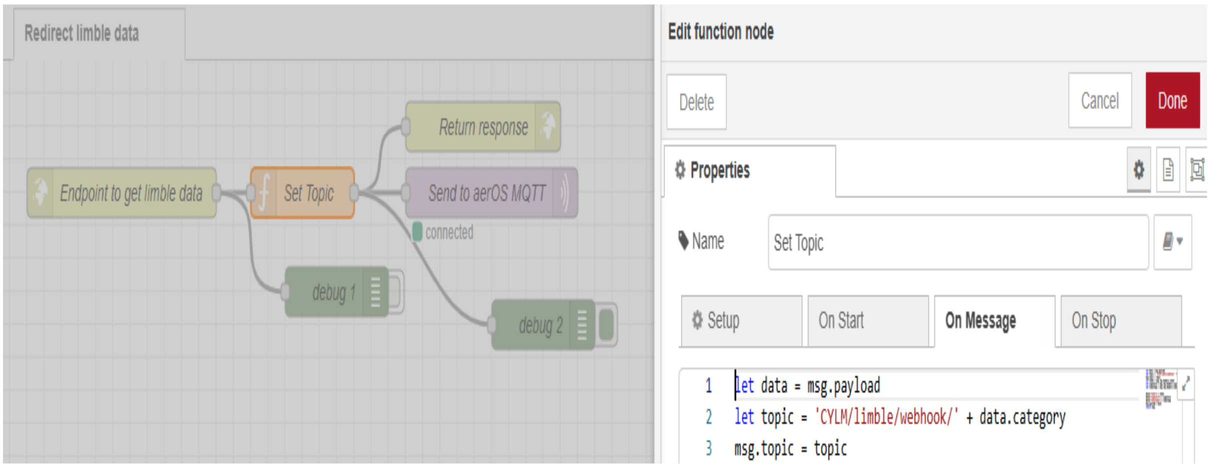
Figure 171. Pilot 4 Straddle carrier sensors and GPS data acquisition.

Pilot4 - Business Process 1 - Activity - 12 (P4-BP1-DA3) TOS data acquisition and integration:

TOPX is the Terminal Operating System used by EUROGATE. It aims at increasing efficiency of the terminal by providing among others integrated solutions for crane planning, as well as real-time automated vessel planning. Thus, different operational information data, such as number of daily working hours per CHE, total weighed containers, terminal block location under work can be obtained. However, despite the original plan was to develop a custom API request (or a recursive SQL query) for the pilot, it was not possible due to the TOPS supplier restrictions, which did not permit access to their services and databases as expected. Hence, since this activity was not mandatory for the successful execution of the pilot, it was finally discarded from the list of work.

Pilot4 - Business Process 1 - Activity - 13 (P4-BP1-DA4) CMMS data acquisition:

Like TOS data, periodic queries to the CMMS system of EUROGATE, namely Limble, was used for correlating the AI-based predictive maintenance models with the actual maintenance tasks. A custom API endpoint in the form of a webhook was developed, which further forwards the maintenance tasks updates to the same MQTT broker, and consequently to the same noSQL database of the pilot.



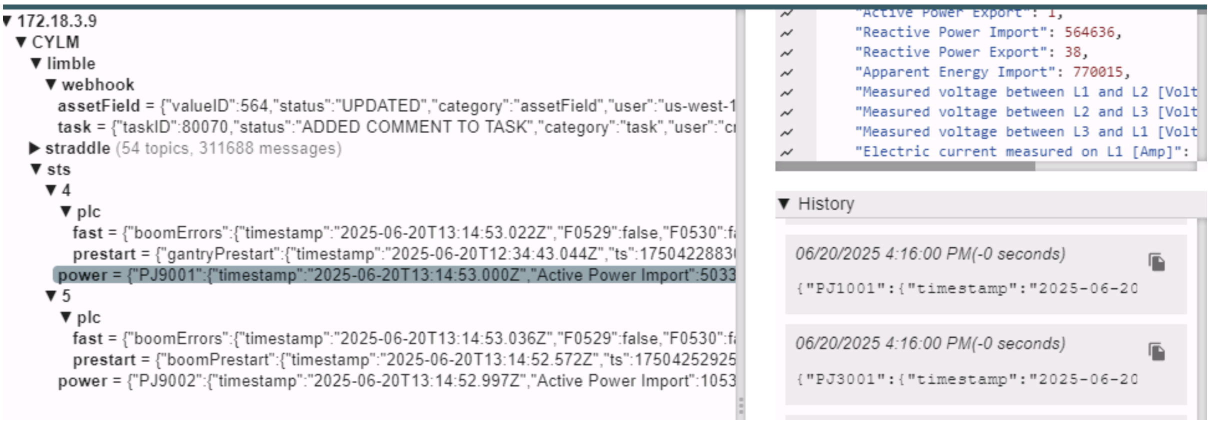


Figure 172.CMMS data acquisition process flow (from Limble webhook, and MQTT transmission).

Pilot4 - Business Process 1 - Activity - 14 (P4-BP1-DA5) STS cranes AI models training:

Different AI models have been trained in order to provide the most accurate predictions for CHEs maintenance. Unfortunately, despite different envisioned AI models from D2.2, only the Trolley wire rope elongation was successfully generated due to the lack of relevant parameters associated with the other use cases.

The background of the use case is as follows. During normal operation of an STS crane, when the trolley component has accumulated several working hours, its wire rope starts elongating. This causes a discrepancy between the distance the limit switches are actually positioned on the travel length of the boom with respect to the first calibration of the machine. When this discrepancy reaches more than 500mm of error, due to safety reasons, the trolley stops operating at nominal speed. Consequently, the technicians then are required to retighten the wire rope by cutting the excess that remains. In order to avoid this situation an early warning system that monitors this discrepancy and predicts when it will become faulty is being put in place, allowing the technical engineering department to plan for this issue ahead of time. To perform the predictions, a classification AI model has been implemented, which compares the current values of the trolley distance between the calibrated distance that were measured during the commissioning, and the corresponding Limit Switches captured from the SL1520.3 register from the PLC of the crane. An example of the trolley wire rope elongation during 4 months of monitoring is illustrated in the below figure.

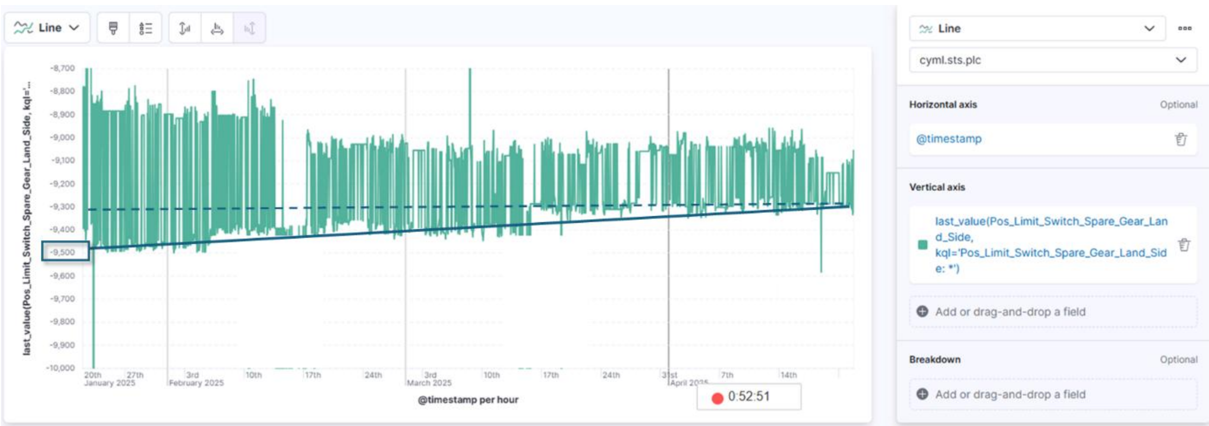


Figure 173. 200mm trolley wire rope elongation obtained by reviewing the spare gear limit switch position stored in the SL1520.3 register of the PLC of the STS crane 04.

Pilot4 - Business Process 1 - Activity - 15 (P4-BP1-DA6) Straddle carriers AI models training:

Different AI models have been also trained for straddle carrier maintenance predictions. In particular, (i) hydraulic system failure, and (ii) engine, brake and inverter overheating predictions were considered good enough to put into operations.

- *Hydraulic system failures*:

The hydraulic system of every industrial system utilizes pumps that work in bursts to maintain the system pressure over a given threshold. Under the usual operation of the Straddle Carriers, the hydraulic pump activates when the system pressure is under a lower threshold and runs continuously until an upper threshold is reached. This creates a roughly inverted sawtooth profile in the pressure-time curve with a period of around 60 seconds when the Straddle Carrier is idle. Naturally, the frequency of those peaks is highly dependent on the behavior at a given moment. That is, some tasks require a higher-pressure demand (such as operating the spreader) and cause the pump to fire more frequently. Some events, such as a leak or a clogged fitting, can have a noticeable impact on the pressure of the whole system or the duty cycle of the pump, e.g., requesting pressure continuously to replace the air lost due to the leak. Thus, by monitoring the duty cycle of the pump (which under normal operation only fires in pulses), some of those issues can be detected before causing significant damage.

The anomaly detection is carried out using a custom variation of the usual Kolmogorov-Smirnov test using the 95th percentile instead of the supremum for increased robustness in the presence of extreme values in long-tailed distributions. This approach, based on traditional statistical methods, was deemed more suitable due to the lack of appropriate labels to train a supervised ML model. Instead, pressure signals are compared against signals from periods where machine operation is verifiably correct (behavior characterization). After a simple clean-up and formatting of the data, each observation is binned in different states according to which subset of the hydraulic actuators is in use. The three actuators of interest are the pressure measurement (PM) for the spreader (PM7), the parking brake (PM33), and the service brakes (PM27). Binary combinations of these actuators being pressurized or not result in 8 different states. Although some of those states do not occur (or rarely do so), such as those involving using the service brakes while the parking brake is engaged (states 1 and 5), only states 0, 2, 3, and 6 occur regularly. Furthermore, observations close to changes in state are discarded, as shown by the different widths of the colour bar in the lower part of the below figure. Instances where a state is held for a time shorter than twice the padding are completely ignored. This is done to obtain cleaner separation of stable states, avoiding boundary conditions.

State	Spreader Engaged	Parking Brake Engaged	Brake Engaged
0	✗	✗	✗
1	✗	✗	✓
2	✗	✓	✗
3	✗	✓	✓
4	✓	✗	✗
5	✓	✗	✓
6	✓	✓	✗
7	✓	✓	✓

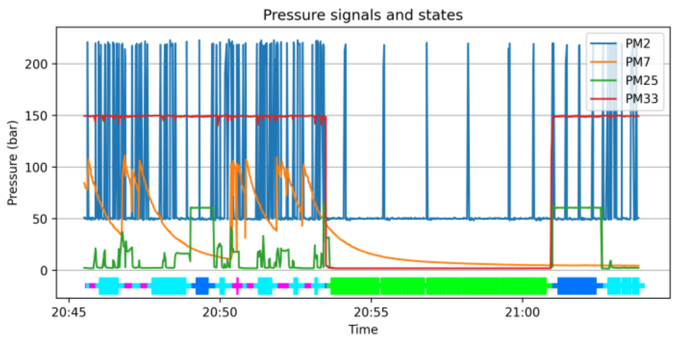


Figure 174. Plot of the pressure measurement at the pump's output (PM2), the three hydraulic signals used to determine the state of the crane (PM7, PM25, and PM33), and the identified state in the color bar below. On the right is a table detailing the specific combination of the three signals that define each of the eight states.

By segregating the behavior into these states, we can expect the duty cycle of the pump to be rather homogeneous in terms of pressure demand. This way, any relevant deviations are indicative of anomalies in the system. These deviations can be observed by comparing a few key statistics of the hydraulic signals against a reference distribution computed for each state from the periods of time where the crane is guaranteed to be in working order. For this study, we have focused on the period between consecutive peaks. Figure illustrates how the distribution for the period between consecutive peaks of the pressure measurement at the pump's output (PM2) is quite different for different states (behaviors) and fairly consistent within each state.

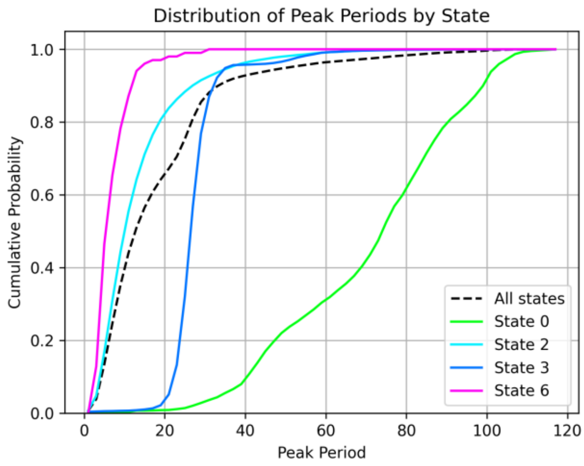


Figure 175. Cumulative distribution of the period between consecutive peaks under regular operating conditions segregated by states. Only common states are shown.

This comparison is performed by computing the residuals between the quantiles of reference and those of the distribution in the most recent hour for each of the states and considering different quantiles of the absolute value of the residuals. The two estimators used to flag anomalies are computed as $q_{0.1}$ and $q_{0.9}$ from the distribution of

$$\{|F_{ref}(x) - F_{obs}(x)|: x \in [0,1]\}$$

Higher values of either of these estimators indicate higher deviations in the observed distribution for the most recent period as compared to the reference distribution under regular operation. However, given the reduced size of the hand-labelled dataset and its unbalanced nature, a decision rule based exclusively on $q_{0.9}$ was chosen to avoid overfitting the criterion. Thus, an observation is considered to be anomalous if $q_{0.9}$ is greater than some threshold determined with the aid of a ROC curve. The threshold value 1.5 was found to provide the best (lower) False Positive Rate (precision) without compromising True positive rate (1 - sensibility) which is considered crucial in order to miss as few true positives as possible.

- *Engine, brake and inverter overheating*:

In this case, we focused on predicting inverter overtemperature faults in SCs, a critical fault that occurred six times during the year 2024. These faults directly affect the cooling system of the Straddle Carriers, leading to elevated inverter or engine temperatures. If left unaddressed, this issue can escalate, potentially resulting in a complete shutdown of the engine or the entire CHE, thereby causing operational delays and safety risks. To develop and evaluate our ML models for inverter overtemperature fault prediction, we selected four of the six recorded failure incidents for training purposes, while the remaining two incidents were reserved for testing. This setup enables the model to learn from historical failure patterns and assess its ability to generalize to unseen events.

Over 200 measurements were recorded every few minutes of operation per SC, including inverter, motor, and engine temperatures, speed, torque, hydraulic pressure, and various error flags. Ambient temperature readings were also recorded using an on-site weather station and incorporated into the dataset. The dataset was manually labelled to consist of two classes: normal and faulty. Faulty data were identified from SCs involved in the recorded incidents, while data from other SCs operating simultaneously were labelled as normal. In total, six inverter overtemperature fault incidents were identified between January and December 2024. The four incidents were used for training, and the two for testing to ensure that the models are tested with incidents they had never observed before. The training dataset size is 8.1 MB and contains 20,210 records (13,860 normal and 6350 faulty), while the testing dataset size is 2.1 MB and includes 5192 records (3947 normal and 1245 faulty). During data preprocessing, missing values were filled using the mean of each metric, one-hot encoding was used for categorical variables, and all numerical variables were standardized by removing the mean and scaling to unit variance.

To perform predictive maintenance utilizing historical data, five popular machine learning models have been evaluated: Artificial Neural Network (ANN), Decision Tree (DT), Random Forest (RF), Extreme Gradient

Boosting (XGBoost), and Gaussian Naive Bayes (GNB). Feature importance was determined by training each model and computing feature importance using the recommended approach for each model. Specifically, we computed permutation importance for ANN, impurity-based importance for DT, RF, and XGBoost, and the log of the coefficients for GNB. Next, we calculated the average importance of each input variable and kept all features with a non-zero importance. In addition, SHAP (SHapley Additive exPlanations) analysis was applied to all models to interpret the contribution of each feature to the model's predictions. Finally, the model hyperparameters were optimized using a 5-fold cross-validated grid search over a parameter grid. The performance of the models for fault prediction was evaluated using accuracy, precision, recall, and F1-score.

Table 21 lists the 21 selected features along with the computed average feature importance for the inverter overtemperature fault prediction. The most important features include the hoist (H) and drive (D) inverter and motor temperatures, which are directly impacted by this type of inverter failure. The *AmbientTemperature* is also a good feature to have as it helps the model differentiate cases of increased inverter temperatures due to weather conditions as opposed to an inverter fault.

Table 21. Feature importance for inverter overtemperature fault prediction.

Feature	Average	Feature	Average
H1InverterTemperature	0.612	***D4InverterTemperature***	0.184
D1MotorTemperature	0.468	***Hoist2SpeedReference***	0.179
D2InverterTemperature	0.448	***Hoist1SpeedReference***	0.179
H2InverterTemperature	0.341	***D3MotorTemperature***	0.179
D3InverterTemperature	0.229	***H2MotorTemperature***	0.176
H1MotorTemperature	0.203	***D2MotorTemperature***	0.158
Hoist2TorqueReference	0.200	***EngineErrorSlot1***	0.158
AmbientTemperature	0.198	***EngineErrorSlot5***	0.154
D1InverterTemperature	0.195	***EngineTemperature***	0.144
Hoist1TorqueReference	0.194	***D4MotorTemperature***	0.136
EngineErrorSlot3	0.193		

To further investigate the impact of each input feature on the output of our predictive maintenance model, we employ the SHAP analysis and visualize the results from the ANN model in the SHAP violin plot in Figure 6. The x-axis shows the SHAP value, which quantifies how much each feature contributes to increasing or decreasing the predicted risk of failure. Positive SHAP values indicate a push toward predicting failure, while negative values suggest a reduced risk. The y-axis lists the features, ranked from top (most impactful) to bottom (least impactful). The color gradient represents the actual value of the feature, with blue indicating low values and red representing high values.

The SHAP violin plot in Figure 6 reveals that the model relies heavily on motor and inverter temperature features, with *H2InverterTemperature*, *H1MotorTemperature*, and *H3InverterTemperature* contributing the most to predictions (ranked high as well). The widespread in SHAP values, especially for these features, suggests strong nonlinear relationships with the target variable. These variables significantly influence the decision boundary and are critical for fault detection or condition classification in this domain.

Other features like *AmbientTemperature* and *EngineErrorSlot3* show moderate influence, while torque and speed references appear to have a smaller impact. These insights help highlight which operational conditions most affect the model's predictions.

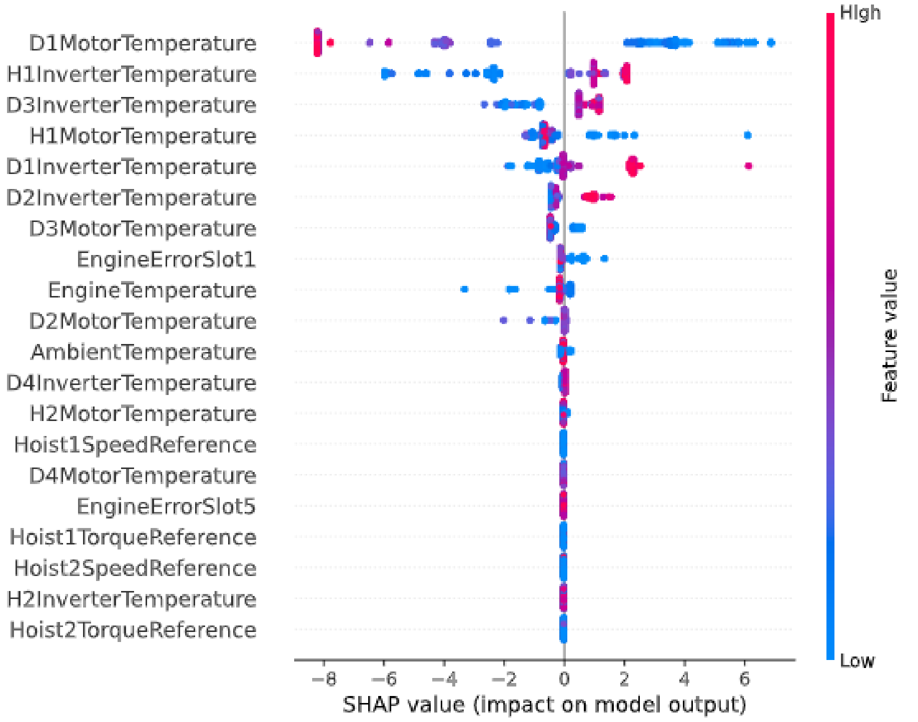


Figure 176. SHAP violin summary plot showing the distribution of feature contributions to the predictive maintenance.

Pilot4 - Business Process 1 - Activity - 16 (P4-BP1-DA7) GIS cartography generation:

The generation of a web-based cartography is considered as extremely relevant for the graphical interface of the pilot. EUROGATE already possesses a digital geo-referenced cartography of the terminal, so the GIS cartography generation activity refers to the export of the AUTOCAD information into a PostGIS compliant component, which is connected to the different data collected from the assets in the terminal.

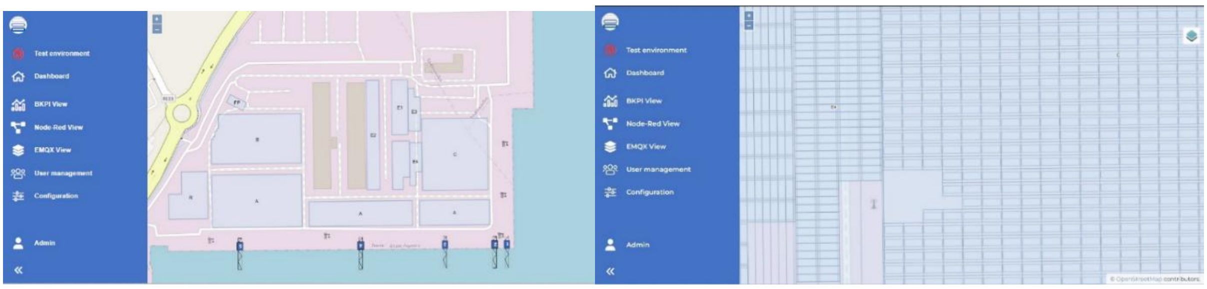


Figure 177. Pilot 4 EUROGATE GIS cartography generation.

2.4.1.1.3. Integration activities

Pilot4 – Business Process 1 – Activity - 17 (P4-BP1-IA1) aerOS core services integration into entrypoint domain (IE0):

The aerOS core services installed in the entrypoint domain of the pilot are: Orion-LD Context Broker, aerOS Federator, Identity manager (Keycloak and OpenLDAP), API Gateway (KrakenD), Self-* suite (self-awareness, self-orchestrator, and self-security), Management Portal (including the benchmark tool), HLO, Red Panda, LLO, and IOTA.

Figure 178. aerOS pods deployed in Pilot 4 entrypoint domain.

Pilot4 – Business Process 1 – Activity - 18 (P4-BP1-IA2) aerOS core services integration into EUROGATE domain (IE5):

The aerOS core services installed in the EUROGATE domain of the pilot are Orion-LD Context Broker, aerOS Federator, API Gateway (KrakenD), Self-awareness, HLO, and LLO.

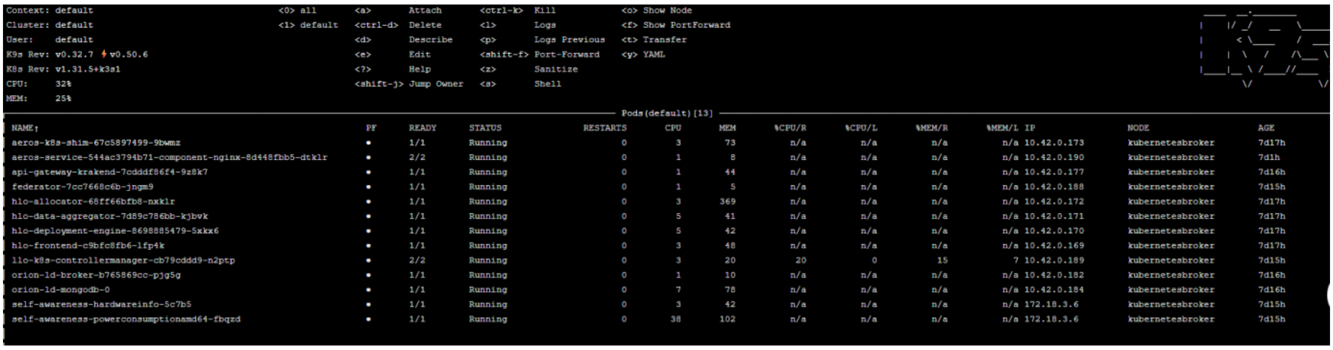


Figure 179. aerOS pods deployed in Pilot 4 EUROGATE domain (IE5).

Pilot4 – Business Process 1 – Activity - 19 (P4-BP1-IA3) aerOS core services integration into EUROGATE domain (IE1-IE4):

The aerOS core services installed in the rest of IEs of EUROGATE domain are only the self-awareness and self-orchestrator for the sts-004 and sts-005. In However, since the IoTs of the Straddle carriers are connected to the aerOS system by means of public 4G network, they have been appended to the network not as k3s nodes, but through the docker-compose elements of the self-* suite as well as the LLO. In addition, in order to bypass the public 4G operator CG-NAT configuration, an overlaid wireguard VPN was also configured, in which the IE5 acted as the wireguard server, while the others were connected as wireguard-peers.

Id	Description	Public Url	Owner	Entrypoint	Status
eurogate_limassol	EUROGATE CTL domain	http://10.10.11.6:31583	pilot4	×	Functional

Infrastructure elements:

Hostname	Container Technology	CPU arch	CPU cores	RAM capacity (MB)	Trust score	Status
kubernetesbroker	Kubernetes	x64	4	16771	-1	Ready
iot2050-debian	Docker	arm64	4	2035	-1	Ready
ie-sch173	Kubernetes	arm64	4	2035	-1	Ready
sts-004	Kubernetes	arm64	4	2035	-1	Ready
sts-005	Kubernetes	arm64	4	2035	-1	Ready

Figure 180. Screenshot of the domains page of the management portal where the IE1-IE4 infrastructure elements of Pilot 4 EUROGATE domain (iot2050 Debian, ie-sch173, sts-004, sts-005).

Pilot4 – Business Process 1 – Activity - 20 (P4-BP1-IA4) aerOS auxiliary services integration into entrypoint domain (IE0):

From the auxiliary services list, Pilot 4 has only deployed the Embedded Analytics Tool in order to provide data drifts insights of the pilot 4 continuum. The following screenshots proof its deployment and use within the pilot environment.

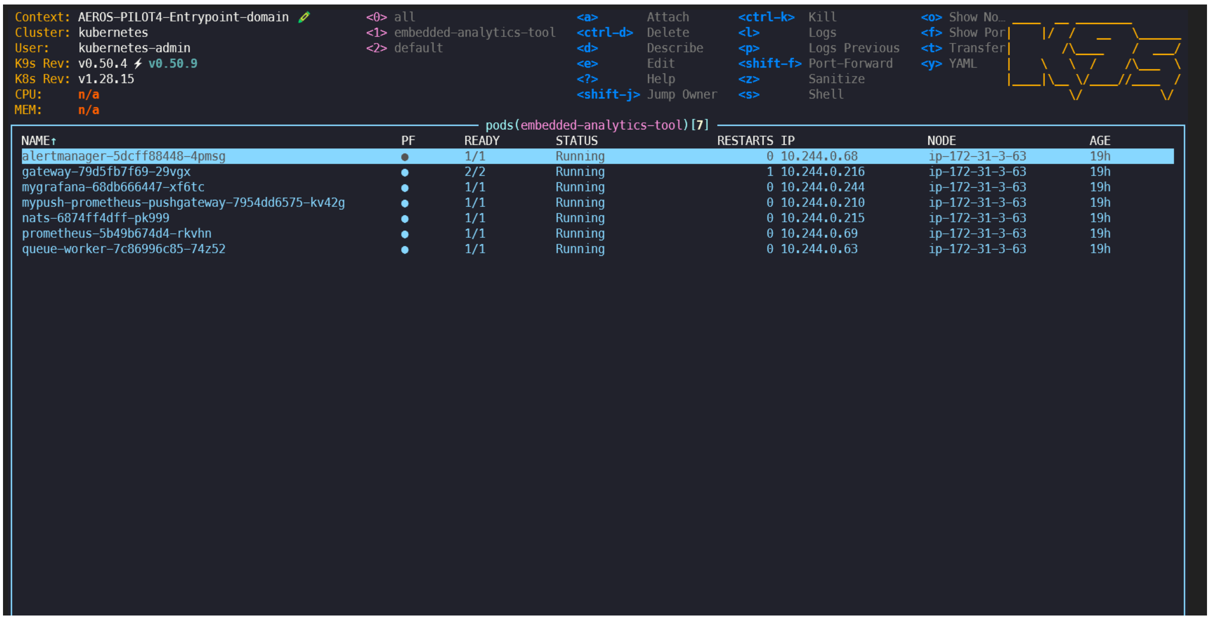


Figure 181. EAT suite deployed in Pilot 4 entrypoint domain.

2.4.2. Risk prevention via Computer Vision in the edge

The second scenario, **Risk prevention via Computer Vision in the edge** goal, was to develop and validate different Computer Vision AI-based algorithms on the edge, allowing EUROGATE to automatically identify containers with damage, and to check for the existence of container seals without the need of human intervention.

2.4.2.1. Technical Schema

Refer to the previous section, where information regarding both pilot scenarios is given.

2.4.2.1. Report of activities

2.4.2.1.1. Setup and procurement activities

Pilot4 – Business Process 2 – Activity - 25 (P4-BP2-SA1) EUROGATE Cameras:

3 IP cameras (HIKVISION DS-2XM6756G0-IS/ND) are installed on every STS crane at the gantry area. They capture the loading/unloading work instructions to/from ships from/to straddle carriers through STS cranes.

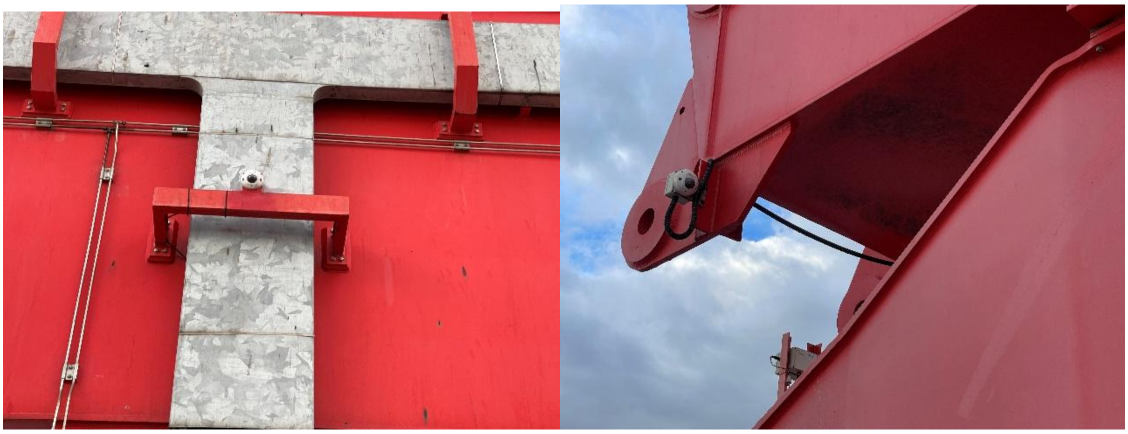


Figure 182. Pilot 4 IP Cameras installed at the STS crane gantry legs

Pilot4 – Business Process 2 – Activity - 26 (P4-BP2-SA2) EUROGATE NVR:

Although the scenario performs the inference of the Computer Vision AI Models on the edge IEs explained below, in the first two years of the project, a large set of videos had to be collected for training and refinement. To do so, a Network Video Recorder was needed. EUROGATE used the Hikvision mobile NVR with 4TB drives, which allowed to connect up to 4-ch IP cameras connectable via PoE interfaces, as well as 4-ch IP cameras extendable via PoE switch. The NVR was able to record up to 5 MP resolution of H.264/H.265 compression for each IP camera.

Figure 183. EUROGATE NVR connected to the IPTV cameras of the pilot.

Pilot4 – Business Process 2 – Activity - 27 (P4-BP2-SA3) Jetson Orin (IE6-IE7):

As mentioned above, the final execution of the different CV models developed in the pilot were inferenced in real time at the edge. To do so, two NVIDIA Jetson AGX Orin Developer Kits with 2048-core NVIDIA Ampere architecture GPU with 64 Tensor cores, 12-core Arm Cortex-A78AE v8.2 64-bit CPU were installed. The development Kit embeds a Linux Board with Ubuntu, as well as CUDA accelerated AI stack, with a set of libraries for acceleration GPU computing and Computer Vision.

Figure 184. Pilot 4 Jetson Orin

Pilot4 – Business Process 2 – Activity – 28 (P4-BP2-SA4) CUT Domain Server (IE8):

The data acquired from the different Pilot 4 domains, and subject of ML development is not only stored at EUROGATE domain, but also at CUT domain for high availability purposes. To do so, a Dell PowerEdge R6525, equipped with 48 AMD physical cores 2.3GHz-3.2GHz turbo, 256MB L3 Cache, 128GB DDR4 RAM, 6 HDDs (10K SAS 1.2 TB each), and an NVIDIA Tesla T4 16GB GPU is connected to the system.

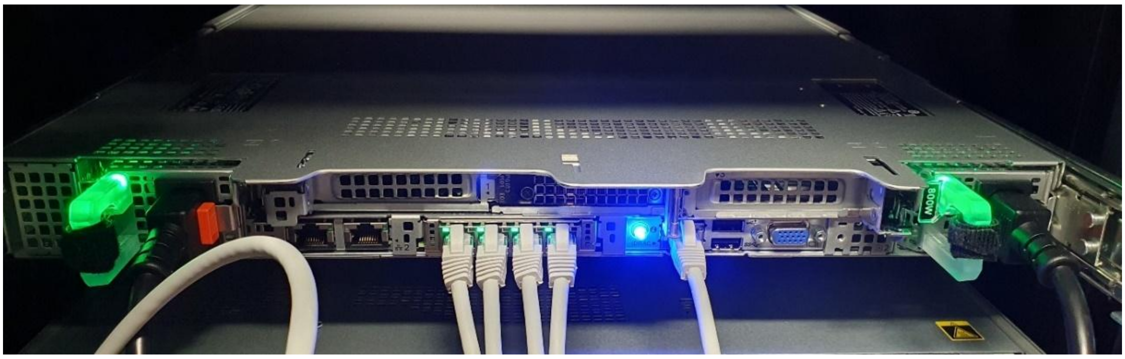


Figure 185. Pilot 4 CUT server of CUT domain.

2.4.2.1.2. Development activities

Pilot4 - Business Process 2 - Activity - 29 (P4-BP2-DA1) Video collection:

A large number of video files were collected in .mp4 format from the EUROGATE NVR for a period of 20 weeks (24/09/2023-24/10/2023, 24/11/2023-24/12/2023, 24/01/2024-24/02/2024, 03/06/2024-30/06/2024, 01/10/2024-27/10/2024). The video files contain both loading and unloading operations of containers from two IP cameras located on STS 04 capturing a variety of environmental conditions (e.g., clear days, misty weather, raining) and lighting conditions (e.g., morning, mid-day, evening, and night). After removing video files that did not record any crane operations, the dataset contains around 16000 video files with a total size of 6.2 TB.

```
$ ls QC4_Gantry_Cam_2
2023_w38_24.09_to_01.10  2023_w47_01.12_to_08.12  2024_w06_08.02_to_14.02  2024_w26_24.06_to_30.06
2023_w39_02.10_to_09.10  2023_w48_09.12_to_16.12  2024_w07_15.02_to_24.02  2024_w40_01.10_to_06.10
2023_w40_10.10_to_17.10  2023_w49_17.12_to_24.12  2024_w23_03.06_to_09.06  2024_w41_07.10_to_13.10
2023_w41_18.10_to_24.10  2024_w04_24.01_to_31.01  2024_w24_10.06_to_16.06  2024_w42_14.10_to_20.10
2023_w46_24.11_to_30.11  2024_w05_01.02_to_07.02  2024_w25_17.06_to_23.06  2024_w43_21.10_to_27.10
$ ls -1 QC4_Gantry_Cam_2/2024_w43_21.10_to_27.10/*.mp4 | head -n 10
QC4_Gantry_Cam_2/2024_w43_21.10_to_27.10/QC4 Gantry Cam2_Crane 4_20241021000001_20241021000206_73437787.mp4
QC4_Gantry_Cam_2/2024_w43_21.10_to_27.10/QC4 Gantry Cam2_Crane 4_20241021001221_20241021001704_73500009.mp4
QC4_Gantry_Cam_2/2024_w43_21.10_to_27.10/QC4 Gantry Cam2_Crane 4_20241021001704_20241021002146_73524962.mp4
QC4_Gantry_Cam_2/2024_w43_21.10_to_27.10/QC4 Gantry Cam2_Crane 4_20241021002146_20241021002627_73549576.mp4
QC4_Gantry_Cam_2/2024_w43_21.10_to_27.10/QC4 Gantry Cam2_Crane 4_20241021003421_20241021003912_73598266.mp4
QC4_Gantry_Cam_2/2024_w43_21.10_to_27.10/QC4 Gantry Cam2_Crane 4_20241021003912_20241021004357_73622277.mp4
QC4_Gantry_Cam_2/2024_w43_21.10_to_27.10/QC4 Gantry Cam2_Crane 4_20241021004357_20241021004853_73646382.mp4
QC4_Gantry_Cam_2/2024_w43_21.10_to_27.10/QC4 Gantry Cam2_Crane 4_20241021004853_20241021005356_73670554.mp4
QC4_Gantry_Cam_2/2024_w43_21.10_to_27.10/QC4 Gantry Cam2_Crane 4_20241021005356_20241021005844_73695126.mp4
QC4_Gantry_Cam_2/2024_w43_21.10_to_27.10/QC4 Gantry Cam2_Crane 4_20241021005844_20241021010320_73719646.mp4
```

Figure 1: Partial listing of collected video files

Pilot4 - Business Process 2 - Activity - 30 (P4-BP2-DA2) Container ID recognition model training:

First, one specialized deep learning model for detecting the container ID text that is located on the container was developed. **The YOLOv12 model was selected**, a state-of-the-art object detection algorithm designed for fast and accurate real-time applications. The model was fine-tuned using the transfer learning methodology with a real, manually labelled dataset that was created from the collected video files. **A total of 1362 images with container IDs were labelled**; some images had IDs in a horizontal form while others had IDs in a vertical form. When the ID appears in a vertical form, a special algorithm was developed to detect each character and then append the characters to form an image containing horizontal text. This process was necessary because almost all current optical character recognition (OCR) tools fail to detect vertical texts accurately. Next, an easyOCR model (a popular python module for extracting text from image) was fine-tuned, also using the transfer learning methodology.

Figure 186. A sample labelled image with both horizontal and vertical container ID.

Pilot4 - Business Process 2 - Activity - 31 (P4-BP2-DA3) Container damage recognition model training:

There were developed two specialized **YOLOv12-based object detection models**: one for identifying containers in motion and another for detecting structural damages on the detected containers. The models were then fine-tuned using the transfer learning methodology with real, manually labelled datasets. **A total of 1927 images with containers were labelled**, containing containers of different sizes (e.g., 20 ft, 40 ft), types (e.g., regular, reefer, open top), and colors (e.g., yellow, blue, red). For the damage detection model, 732 images with damage were labelled. Including bents, dents, holes, and normal wear. Data augmentation techniques were used to expand our datasets to 3854 and 1764 images, for container and damage, respectively. The training process involved fine-tuning several key training parameters to optimize performance, including the hue and saturation of the input images as well as tuning the IoU threshold to better detect multiple containers and damages that appear close together.

Figure 187: Sample of labelled images from (a) the container and (b) the damage datasets

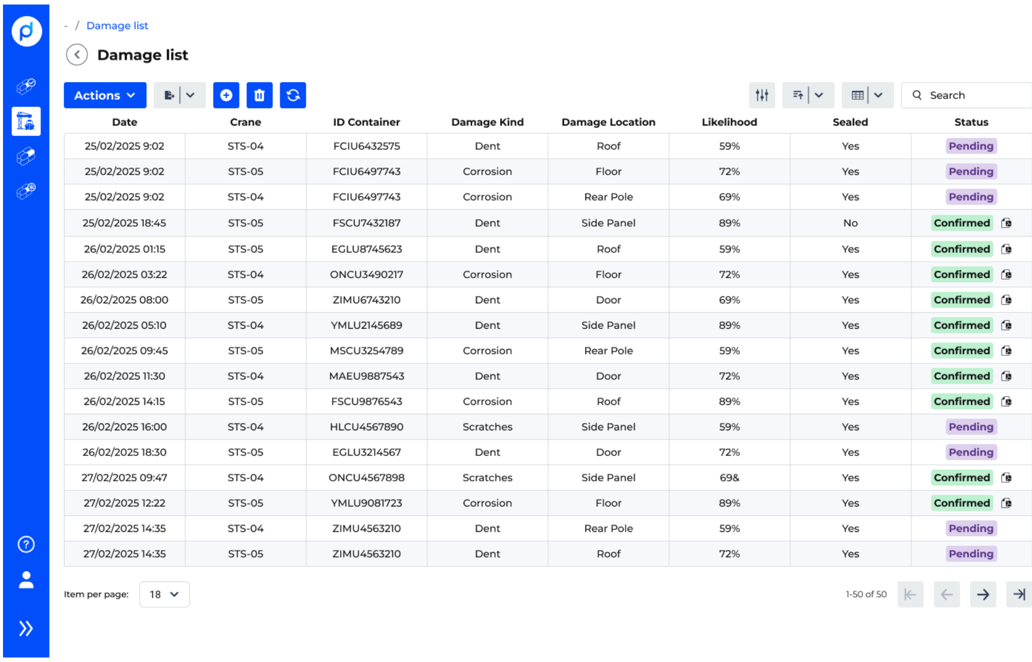
Pilot4 - Business Process 2 - Activity - 32 (P4-BP2-DA4) Container seal recognition model training:

We developed two specialized YOLOv12-based object detection models: one for identifying container doors and another for detecting the presence of seals on the detected container doors. The models were then fine-tuned using the transfer learning methodology with real, manually labelled datasets. Two datasets were created, one containing 300 images of container doors and backs of different colors, and one containing 509 images of container doors with and without seals. The training process for both models involved fine-tuning the same key training parameters discussed above for the container damage recognition models to optimize model performance.

Figure 188. Sample of labelled images from (a) the container door and (b) the seal datasets.

Pilot4 - Business Process 2 - Activity - 33 (P4-BP2-DA5) Yard inventory damaged container dashboard:

The outputs generated by the three CV models developed in the previous activities are compiled and forwarded to a web application available to EUROGATE Operations department. This web application contains a form with the list of containers detected as damaged, including ID, date, time, damage type, damage location, likelihood, seal. It also includes the associated autogenerated PDF report that contains the proofs of the damage detected, which operational staff can confirm before reporting to customers. A screenshot of the dashboard is illustrated below.



Date	Crane	ID Container	Damage Kind	Damage Location	Likelihood	Sealed	Status
25/02/2025 9:02	STS-04	FCIU6432575	Dent	Roof	59%	Yes	Pending
25/02/2025 9:02	STS-05	FCIU6497743	Corrosion	Floor	72%	Yes	Pending
25/02/2025 9:02	STS-04	FCIU6497743	Corrosion	Rear Pole	69%	Yes	Pending
25/02/2025 18:45	STS-05	FSCU7432187	Dent	Side Panel	89%	No	Confirmed
26/02/2025 01:15	STS-05	EGLU8745623	Dent	Roof	59%	Yes	Confirmed
26/02/2025 03:22	STS-04	ONCU3490217	Corrosion	Floor	72%	Yes	Confirmed
26/02/2025 08:00	STS-05	ZIMU6743210	Dent	Door	69%	Yes	Confirmed
26/02/2025 05:10	STS-04	YMLU2145689	Dent	Side Panel	89%	Yes	Confirmed
26/02/2025 09:45	STS-05	MSCU3254789	Corrosion	Rear Pole	59%	Yes	Confirmed
26/02/2025 11:30	STS-04	MAEU9887543	Dent	Door	72%	Yes	Confirmed
26/02/2025 14:15	STS-05	FSCU9876543	Corrosion	Roof	89%	Yes	Confirmed
26/02/2025 16:00	STS-04	HLCU4567890	Scratches	Side Panel	59%	Yes	Pending
26/02/2025 18:30	STS-05	EGLU3214567	Dent	Door	72%	Yes	Pending
27/02/2025 09:47	STS-04	ONCU4567898	Scratches	Side Panel	69&	Yes	Confirmed
27/02/2025 12:22	STS-05	YMLU9081723	Corrosion	Floor	89%	Yes	Confirmed
27/02/2025 14:35	STS-04	ZIMU4563210	Dent	Rear Pole	59%	Yes	Pending
27/02/2025 14:35	STS-05	ZIMU4563210	Dent	Roof	72%	Yes	Pending

Figure 189. Damaged container dashboard.

2.4.2.1.3. Integration activities

Pilot4 – Business Process 2 – Activity - 34 (P4-BP2-IA1) aerOS core services integration into CUT domain (IE6):

The aerOS core services installed in the CUT domain of the pilot are Orion-LD Context Broker, aerOS Federator, API Gateway (Kraken-D), Self-awareness, HLO, and LLO.

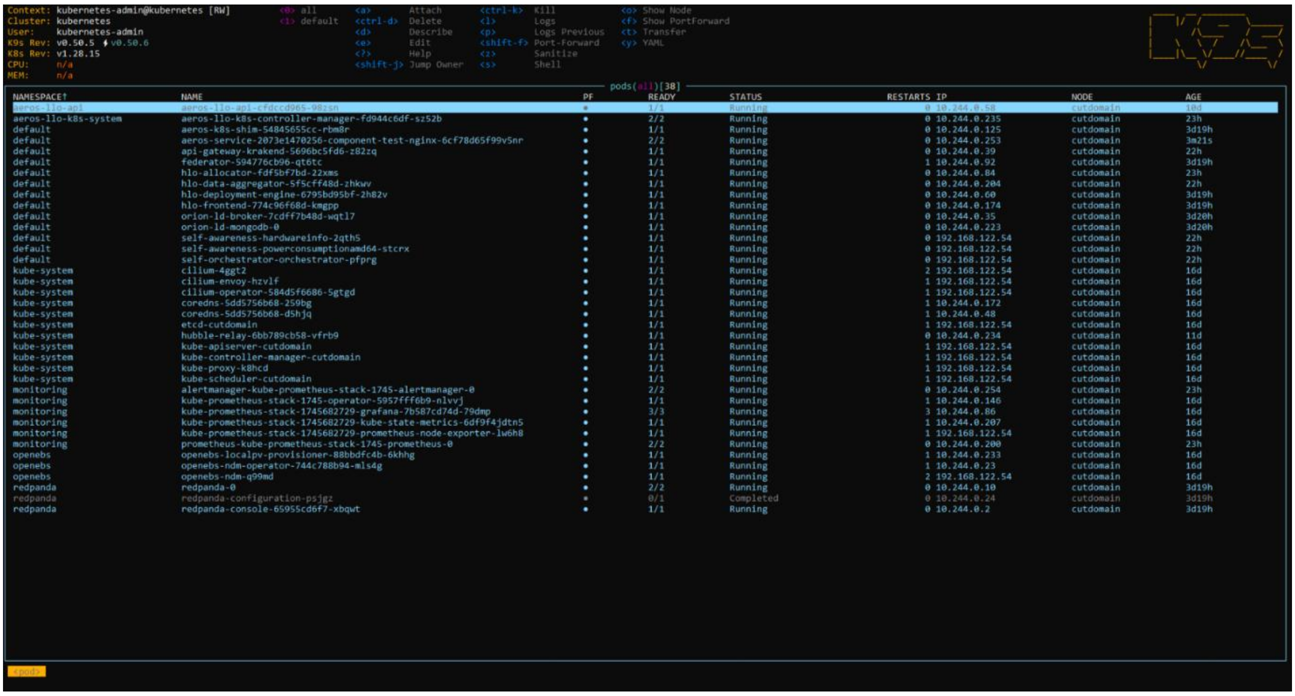


Figure 190. aerOS pods deployed in Pilot 4 CUT domain

The three domains deployed for the pilot are accessible from the management portal, and can be seen in the following screenshot.

Id	Description	Public Url	Owner	Entrypoint
pilot4_domain1	Pilot4 domain1	https://aeros-pilot4.prodevelopataws.com	pilot4	✓
eurogate_limassol	EUROGATE CTL domain	http://172.18.3.6:31583	pilot4	×
cutdomain	This is a new domain, which is not the entrypoint	https://cutdom.aeros.cut.ac.cy	CUT	×

Figure 191. Pilot 4 domains available from the management portal.

Pilot4 – Business Process 2 – Activity - 35 (P4-BP2-IA2) aerOS core services integration into Jetson Orin (IE7-IE8):

Like P4-BP1-IA3 integration activity from previous scenario, the IE7 and IE8 are IEs of EUROGATE domain, so that only the self-awareness and self-orchestrator was needed to integrate them. The following figures show all the IEs of the domain (from management portal domain and continuum tabs), included both Jetson Orin IEs.

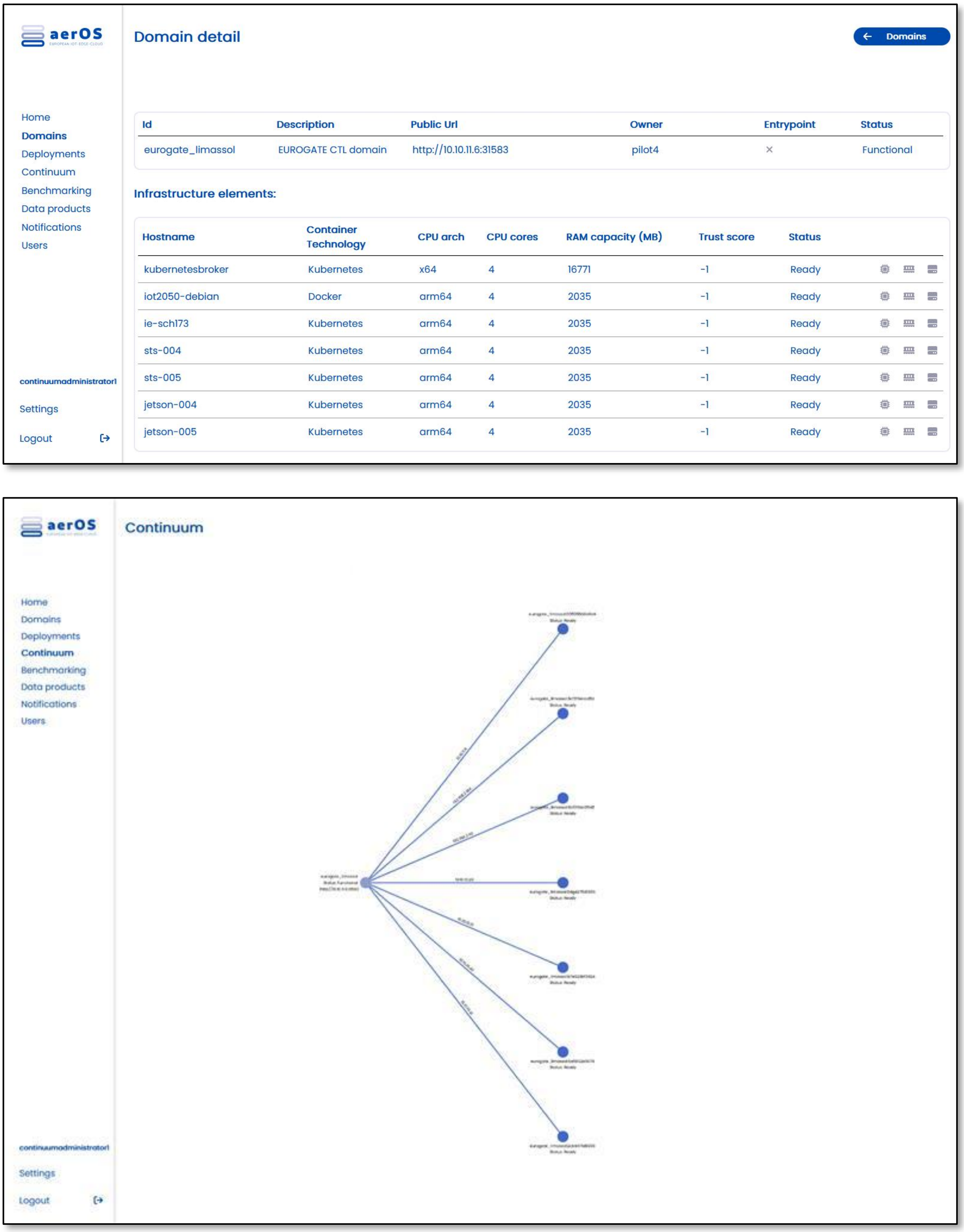


Figure 192. Screenshot of the Domains (top) and Continuum (bottom) pages of the management portal where all the IEs of Pilot 4 EUROGATE domain are integrated, including jetson-004 and jetson-005.

2.4.3. Pilot 4 Time-plan

The following Gantt diagram list all the activities from M19 (February 2024) until M35 (July 2025), representing evolution from status declared in D5.3.

Table 22. Pilot 4 updated Gantt timeline

Pilot 4		2024											2025								
Code	**Name**	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33	M34	M35	M36	M37	M38
Business Process (BP) 1 (Scenario 1) - Predictive maintenance of Container Handling Equipment																					
Setup & Procurement Activities																					
P4-BP1-SA1	A1 - STS Cranes PLCs																				
P4-BP1-SA2	A2 - Straddle Carriers PLCs																				
P4-BP1-SA3	A3 - Straddle Carriers sensors																				
P4-BP1-SA4	A4 - Straddle Carriers GPSs																				
P4-BP1-SA5	A5 - Straddle Carriers Human Machine Interfaces																				
P4-BP1-SA6	A6 - Straddle Carriers 4G Routers																				
P4-BP1-SA7	A7 - Straddle Carriers and STS IoT Gateways (IE1-IE4)																				
P4-BP1-SA8	A8 - EUROGATE Domain Server (IE5)																				
P4-BP1-SA9	A9 - Entrypoint domain Server (IE0)																				
Development Activities																					
P4-BP1-DA1	A10 - PLC data gathering																				
P4-BP1-DA2	A11 - GPS and sensors data gathering																				
P4-BP1-DA3	A12 - TOS data acquisition																				
P4-BP1-DA4	A13 - CMMS data acquisition																				
P4-BP1-DA5	A14 - STS cranes AI models training																				
P4-BP1-DA6	A15 - Straddle carriers AI models training																				
P4-BP1-DA7	A16 - GIS cartography generation																				
Integration Activities																					
P4-BP1-IA1	A17 - aerOS core services integration into etrypoint domain (IE0)																				
P4-BP1-IA2	A18 - aerOS core services integration into EUROGATE domain (IE5)																				
P4-BP1-IA3	A97 - aerOS core services integration into EUROGATE domain																				

	(IE1-IE4)
P4-BP1-IA4	A20 - aerOS auxiliary services integration into entrypoint domain (IE0)
Validation Activities	
P4-BP1-VA1	A21 - Data acquisition
P4-BP1-VA2	A22 - Data storage
P4-BP1-VA3	A23 - STS and Straddle carriers AI model inference verification
P4-BP1-VA4	A24 - aerOS entrypoint domain - EUROGATE domain communication
BP2 (Scenario 2) - Risk prevention via Computer Vision in the edge	
Setup & Procurement Activities	
P4-BP2-SA1	A25 - EUROGATE Cameras
P4-BP2-SA2	A26 - EUROGATE NVR
P4-BP2-SA3	A27 - Jetson Orin (IE6-IE7)
P4-BP2-SA4	A28 - CUT Domain Server (IE8)
Development Activities	
P4-BP2-DA1	A29 - Video collection
P4-BP2-DA2	A30 - Container ID recognition model training
P4-BP2-DA3	A31 - Container damage recognition model training
P4-BP2-DA4	A32 - Container seal recognition model training
P4-BP2-DA5	A33 - Yard inventory damaged container dashboard
Integration Activities	
P4-BP2-IA1	A34 - aerOS core services integration into CUT domain (IE6)
P4-BP2-IA2	A35 - aerOS core services integration into Jetson Orin (IE7-IE8)
Validation Activities	
P4-BP2-VA1	A36 - Video storage
P4-BP2-VA2	A37 - CV model inference verification
P4-BP2-VA3	A38 - aerOS entrypoint domain - CUT domain communication

2.5. Pilot 5 - Energy Efficient, Health Safe & Sustainable Smart Buildings

Pilot 5 has been built to accommodate the aerOS MetaOS architectural concepts to achieve the innovative goals in the Smart Building vertical. The application components have been developed, transformed, and integrated to be deployed in the aerOS continuum through a list of pilot level technical activities that are summarized in the form of a Gantt chart at the end of this section.

The fundamental development effort for Pilot 5 has been to:

- Create the aerOS domains in Pilot 5 including the basic aerOS services already achieved in the first project period (P1),
- Develop and customize the Pilot 5 application components (IoT system, Forecast System, Recommender, GUI) to operate in the aerOS continuum, which is a continuous process addressed through an iterative process (first and final integration cycles) with related technical activities appropriately grouped and listed in in the table below,
- Incorporate the optional aerOS services as well as maintain latest aerOS components' releases, which is also a continuous process. It must be noted that Pilot 5 had been selected to apply and validate the data fabric implementations in the application context (addressing smart buildings ontology), and experiment with semantic interoperability using the Orion-LD context broker and related data catalogues as the means for the application components to interact.

The following table showcases how Pilot 5 has switched from processing data in the Cloud to the Edge:

Table 23. Pilot 5 Edge vs Cloud comparison before and after aerOS.

	Cloud	Edge	Far Edge
Before aerOS	IoT Backend System Forecasting System+AI GUI	IoT Gateway	IoT sensors
After aerOS	-	IoT Back-end System Forecasting System + AI Recommender GUI	IoT Gateway, IoT sensors

2.5.1. Intelligent Occupational Safety and Health

2.5.1.1. Technical Schema

Pilot 5 is operating in two aerOS Domains, that are presented in detail in the following topology diagram:

- Domain01 is the pilot's Entry Domain and contains all aerOS services necessary for the administration of the Pilot continuum (a.k.a. Entry Domain). It contains two Infrastructure Elements (IEs) hosting the Pilot 5 application/IoT backend services.
- Domain02 is the pilot's Main Domain and contains five IEs, including three KubeEdge nodes containing the IoT Edge applications (HomeAssistant).

The Pilot 5 aerOS ontology diagram can be viewed in the following figure.

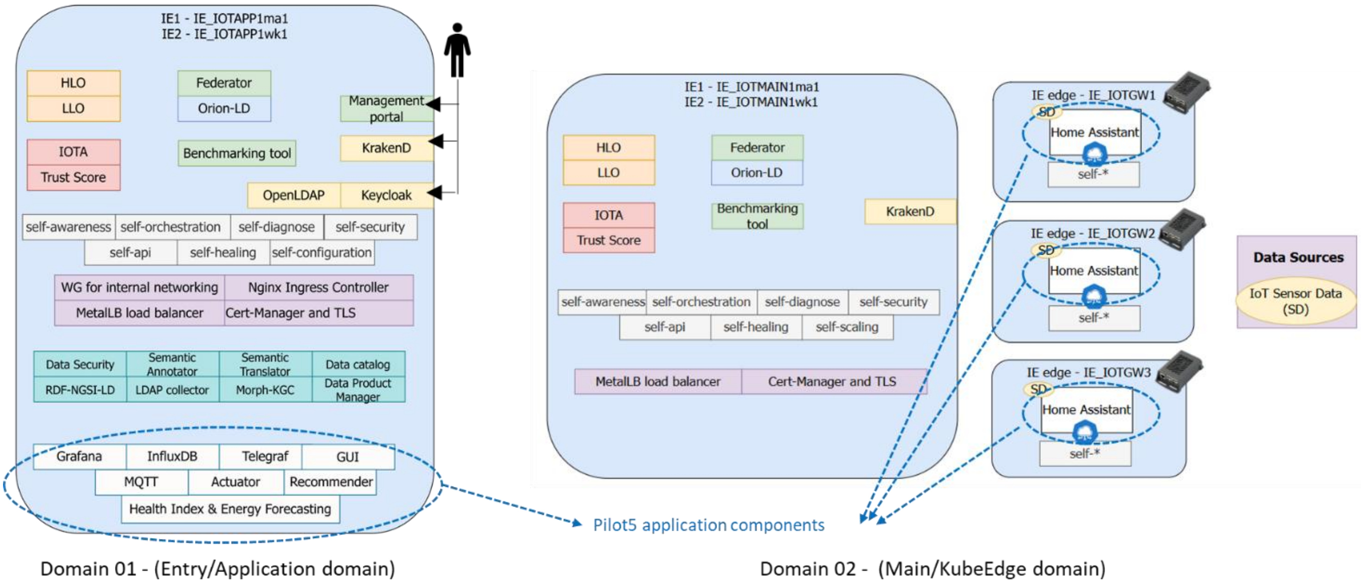


Figure 193. Pilot 5 Topology diagram – Technical schema.

2.5.1.1. Report of activities

2.5.1.1.1. Setup and procurement activities

P5-BP1-SA1 Site surveys for the selection of Pilot 5 Building/Rooms

The selection of the rooms to host the pilot demonstrator as well as the employees that will participate and engage with the application recommendations was completed within Year 1. The physical location concerning the building and rooms can be seen in the following figure.

Figure 194. Pilot 5 Smart Building and Rooms

P5-BP1-SA2 Procurement of Servers & Equipment

All necessary equipment to support the pilot developments have been secured and / or procured. Primarily this relates to the sensors installed in the Pilot 5 rooms. Examples of such equipment can be viewed in the following figures.

Figure 195. Air Purifier in aerOS:

Figure 196. Dehumidifier device in aerOS

P5-BP1-SA3 Identification of Appropriate Smart Building Sensors

As part of the initial setup activities, the id entification of the appropriate sensors to furnish each room has resulted as depicted in following tables for Rooms 105, 106, 208 and 209 respectively.

Table 24. Pilot 5 Room 105 sensors.

Room 105		
device_name	**device_id**	**Metrics**
Environmental Sensor	10504062801	Temp, Humidity, Pressure
Innr switch	10502070602	state(On/Off)
PIR (1)	10506062403	Presence, Motion
Door Sensor	10508062504	Open, Closed
Weather (outdoors)	10527986605	Temp, Humidity, Pressure
PIR (2)	10506062406	Presence, Motion
Window Sensor	10507032007	Open, Closed
Xiaomi Mi Air Purifier 4 Pro	EC:4D:3E:6A:4D:2F	PM2.5, state(On/Off)
Inventor deHumidityfier EVA ION Pro	38:1F:8D:5E:54:C2	Temp, Humidity, state(On/Off)
CO2 tracker 12	SenseAir-CO2-12_12	CO2,PM1,PM2.5,PM10, Temp, Humidity, Pressure
SGP40	08:3A:F2:8C:B4:78	AirQ
MQ135 sensor	84:CC:A8:48:89:F4	ACETONA, ALCOHOL, CO, TOLUENO, NH4
Shelly plug s (1)	10502046917	Power, Energy, state(On/Off)
Power meter total measurements	10501040318	Power, Energy
Shelly plug s (2)	10502046919	Power, Energy, state(On/Off)

Table 25. Pilot 5 Room 106 sensors.

Room 106		
device_name	**device_id**	**Metrics**
Weather (outdoors)	10627986605	Temp, Humidity, Pressure
SGP40	08:3A:F2:B9:85:44	AirQ
Window Sensor	10608054116	Open, Closed
Door Sensor	10608062517	Open, Closed
PIR (1)	10606062418	Presence, Motion

Room 106		
device_name	**device_id**	**Metrics**
Environmental Sensor	10604062819	Temp, Humidity, Pressure
CO2 tracker 6	SenseAir-CO2-6_6	CO2,PM1,PM2.5,PM10, Temp, Humidity, Pressure
PIR (2)	10606054321	Presence, Motion
Shelly plug s (1)	10602046922	Power, Energy, state(On/Off)
Shelly plug s (2)	10602046923	Power, Energy, state(On/Off)
Shelly plug s (3)	10602046924	Power, Energy, state(On/Off)
Power meter	10601040325	Power, Energy
Mq135	48:E7:29:6D:EF:93	ACETONA, ALCOHOL, CO, TOLUENO, NH4

Table 26. Pilot 5 Room 208 sensors.

Room 208		
device_name	**device_id**	**Metrics**
Environmental Sensor	20804062801	Temp, Humidity, Pressure
Innr switch	20802070602	Open, Closed
PIR (1)	20806062403	Presence, Motion
Door Sensor (main)	20808054104	Open, Closed
Weather (outdoors)	20827986605	Temp, Humidity, Pressure
PIR (2)	20806054306	Presence, Motion
Window Sensor	20808054107	Open, Closed
PIR (3)	20806054308	Presence, Motion
PIR (4)	20806054309	Presence, Motion
Door Sensor (balcony)	20808054110	Open, Closed
CO2 tracker 9	SenseAir-CO2-9_9	CO2,PM1,PM2.5,PM10, Temp, Humidity, Pressure
QualitAir HVAC 24BTU	20817267012	Power, Energy, state(On/Off)
Shelly plug s (1)	20802046913	Power, Energy, state(On/Off)
Shelly plug s (2)	20802046914	Power, Energy, state(On/Off)
Power meter	20801040315	Power, Energy
PIR (5) - Whole room	20806062416	Presence, Motion
Mq135	84:0D:8E:A5:3B:46	ACETONA, ALCOHOL, CO, TOLUENO, NH4

Table 27. Pilot 5 Room 209 sensors.

Room 209		
device_name	**device_id**	**Metrics**
Environmental Sensor	20904062801	Temp, Humidity, Pressure
Innr switch	20902070602	Open, Closed
PIR	20906054303	Presence, Motion
Door Sensor (main)	20908054104	Open, Closed
Weather (outdoors)	20927986605	Temp, Humidity, Pressure
Window Sensor	20908054107	Open, Closed

Room 209		
device_name	**device_id**	**Metrics**
HVAC measurements	20917267008	Power, Energy, state(On/Off)
Shelly plug s (1)	20902046909	Power, Energy, state(On/Off)
Power meter (total)	20901040310	Power, Energy
CO2 tracker 5	SenseAir-CO2-5_5	CO2,PM1,PM2.5,PM10, Temp, Humiditydity, Pressure
Mq135	84:0D:8E:A5:7C:D4	ACETONA, ALCOHOL, CO, TOLUENO, NH4

2.5.1.1.2. Development activities

P5-BP1-DA4 Deployment & Maintenance of Sensors

The deployment of the IoT sensors necessary to support the pilot executions have been completed in Year 1 of the project, however continuous maintenance and repair activities are necessary to assure accurate and complete sets of the sensor data.

P5-BP1-DA5 Deployment of the IoT backend/Home Assistant

The installation of the IoT backend to orchestrate the smart building decisions in the Pilot 5 rooms has been completed in Year 1 of the project. An indicative view of the deployment in all 4 rooms can be seen in Figure .

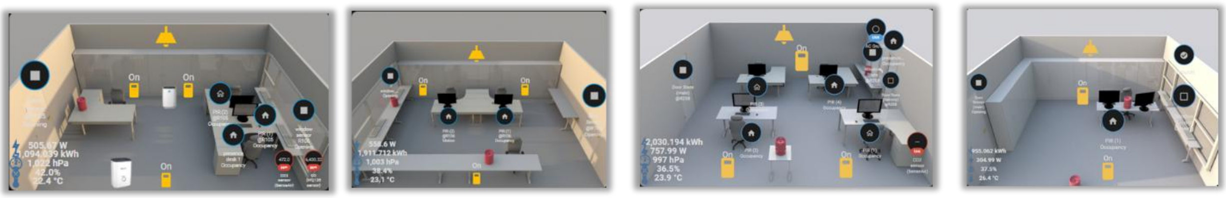


Figure 197. Pilot 5 Home Assistant room and sensors layout

P5-BP1-DA6 Installation of the aerOS Domains

The aerOS domains that are part of the pilot architecture are seen in Figure .

The Installation of the aerOS domain in the infrastructure of Pilot 5 can be seen for the Domain01: Entry Domain and Domain02: Main Domain in Figure and Figure , respectively.

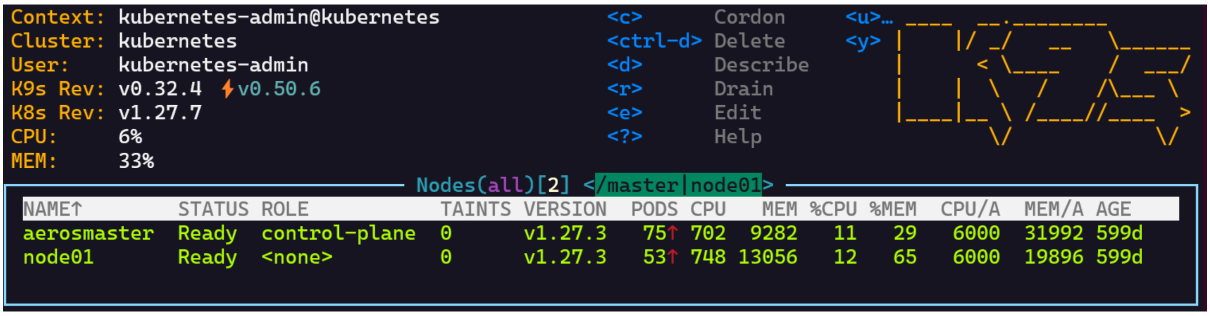


Figure 198. Pilot 5 aerOS Domain01: Entry Domain Installation

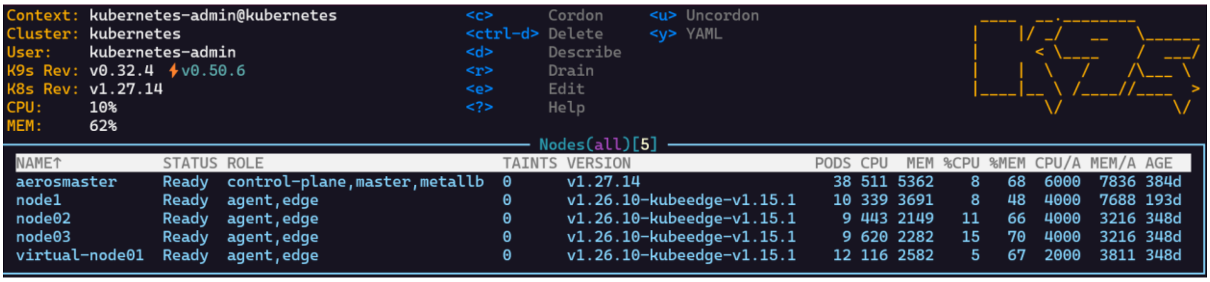


Figure 199. Pilot 5 aerOS Domain02: Main Domain Installation

P5-BP1-DA7 Transformation of the IoT backend as aerOS IE

To adapt to the aerOS meta operating systems, all the Pilot 5 application components must be containerized. The pilot's IoT system was running as a virtualized service and had to be converted respectively. In Figure the application pods running in Domain01 are depicted and in Figure the ones running in Domain02.

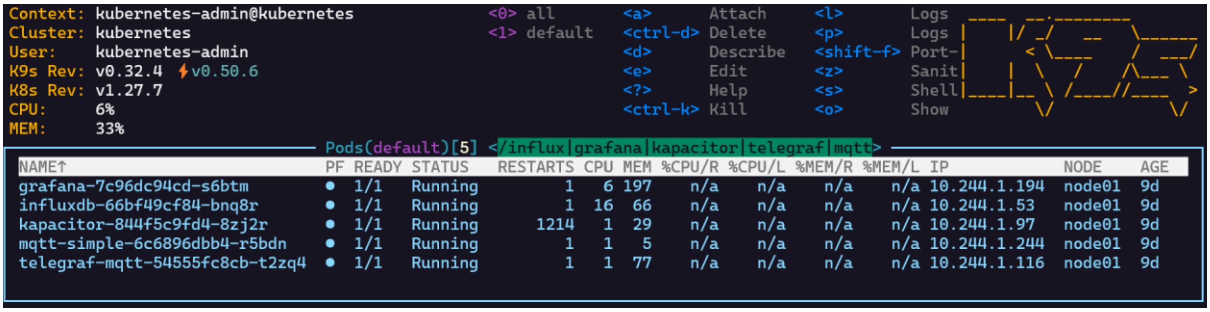


Figure 200. Pilot 5 Containarised IoT system – Running on Domain01

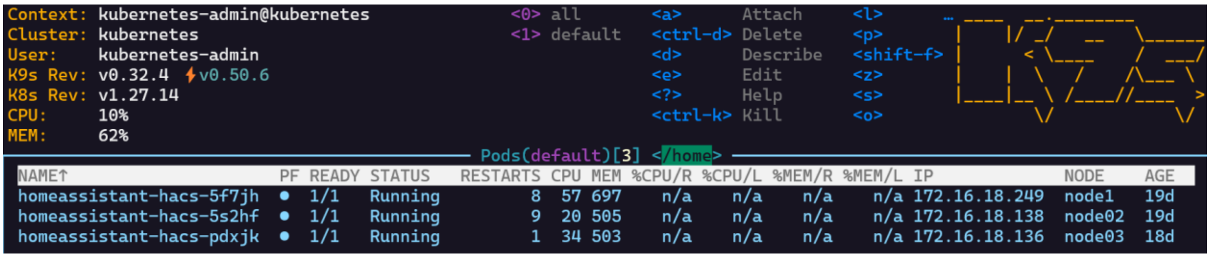


Figure 201: Pilot 5 Containarised IoT system – Running on Domain02

P5-BP1-DA8 HVAC/Plugs Actuator Component

The actuator system receives optimal room condition parameters—such as temperature, humidity, CO_2 levels, and PM2.5 concentration—from the forecaster. Based on this input, it intelligently controls environmental devices like the HVAC system, air purifier, or dehumidifier to bring the room to the desired state. For instance, if CO_2 or PM2.5 levels are high, the air purifier is activated; if humidity is too high, the dehumidifier is turned on; and if the temperature deviates from the optimal range, the HVAC system adjusts to heat or cool accordingly. Once the forecast predicts that environmental conditions will remain within optimal ranges naturally, the actuator receives a signal from the optimization system to power down the devices, ensuring energy efficiency and comfort.

The actuator component is a Python-based controller that listens for room environment optimization notifications on the *Orion_notifications_optimization* topic. Upon receiving a message for a room, it processes environmental targets such as temperature, humidity, CO_2, and PM2.5. A print-screen of the actuator component running is provided in Figure .

```
/app/actuator_controller.py:182: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
🚀 Starting MQTT actuator controller...
✅ Connected to MQTT broker
📡 Subscribed to topic: Orion_notifications_optimization

📨 Received MQTT notification
🔍 Detected change for room R208

🔄 Processing controls for room R208
📊 Received values:
  Temperature target: 20.2°C
  Humidity target: 34.2%
  CO2 target: 350.2ppm
  PM2.5 target: 15.2µg/m³
  Energy optimization: 1.0

📊 Current conditions in R208:
  Outside temperature: 32.4°C
  Room temperature: 28.18°C
  Room humidity: 33.52%
  Room CO2: 434.0ppm
  Room PM2.5: 9.8µg/m³
💧 Humidity at optimal level - turning off dehumidifier
🌬 Air quality needs improvement - turning on air purifier
  CO2: 434.0ppm > 350.2ppm
🌡 Season detected: summer
🌬 Setting AC to cool mode at 20.0°C
✅ Completed processing for room R208


📨 Received MQTT notification
🔍 Detected change for room R208

🔄 Processing controls for room R208
📊 Received values:
  Temperature target: 20.2°C
  Humidity target: 34.2%
  CO2 target: 350.2ppm
  PM2.5 target: 15.2µg/m³
  Energy optimization: 0.0

📊 Current conditions in R208:
  Outside temperature: 32.4°C
  Room temperature: 28.18°C
  Room humidity: 33.52%
  Room CO2: 434.0ppm
  Room PM2.5: 10.8µg/m³
⚠ Energy optimization is 0 - turning off all devices
✅ Air purifier turned off
✅ Dehumidifier turned off
✅ AC turned off
✅ Completed processing for room R208
```

Figure 202. Pilot 5 Actuator Logs

P5-BP1-DA9 Forecast Engine - Health Index AI Component

As part of the development activities in Pilot 5, we implemented the concept of a room health score, a metric that reflects the overall quality of the indoor environment with regard to human comfort and safety. By combining important environmental factors including temperature, humidity, CO_2 levels, and PM into a single easily comprehensible number between 0 and 100, this score contributes to the larger objective of encouraging healthier and more energy-efficient workplaces. In order to arrive at a single score, the computation compares each parameter to literature-based optimal ranges. The health score is determined both in real time and through forecasting, allowing for proactive planning based on anticipated trends and dynamic responses to current circumstances. The health score is kept up to date and stored in the Context Broker (Orion-LD) to facilitate system-wide integration making it easily accessible to every other element of the Pilot 5 ecosystem.

calculate_health_score.py 1.72 KiB

Edit Replace Delete

```python
def calculate_health_score(temp_in, humi_in, co2_in, pm25_in, pm10_in):
    """Calculate the health score based on environmental parameters."""
    # Optimal ranges
    optimal_temp_min, optimal_temp_max = 18, 24
    optimal_humidity_min, optimal_humidity_max = 30, 60
    optimal_co2_max = 1000
    optimal_pm25_max = 12  # µg/m³
    optimal_pm10_max = 20  # µg/m³

    # Temperature score
    if optimal_temp_min <= temp_in <= optimal_temp_max:
        temp_score = 100
    elif temp_in < optimal_temp_min:
        temp_score = max(0, (temp_in - (optimal_temp_min - 5)) / 5 * 100)
    else:
        temp_score = max(0, ((optimal_temp_max + 5) - temp_in) / 5 * 100)

    # Humidity score
    if optimal_humidity_min <= humi_in <= optimal_humidity_max:
        humidity_score = 100
    elif humi_in < optimal_humidity_min:
        humidity_score = max(0, (humi_in - (optimal_humidity_min - 20)) / 20 * 100)
    else:
        humidity_score = max(0, ((optimal_humidity_max + 20) - humi_in) / 20 * 100)

    # CO2 score
    if co2_in <= optimal_co2_max:
        co2_score = 100
    elif co2_in <= 2000:
        co2_score = max(0, (2000 - co2_in) / 1000 * 100)
    else:
        co2_score = 0

    # PM2.5 score
    if pm25_in <= optimal_pm25_max:
        pm25_score = 100
    elif pm25_in <= 35:
        pm25_score = max(0, (35 - pm25_in) / (35 - optimal_pm25_max) * 100)
    else:
        pm25_score = 0

    # PM10 score
    if pm10_in <= optimal_pm10_max:
        pm10_score = 100
    elif pm10_in <= 50:
        pm10_score = max(0, (50 - pm10_in) / (50 - optimal_pm10_max) * 100)
    else:
        pm10_score = 0

    # Average score
    health_score = (temp_score + humidity_score + co2_score + pm25_score + pm10_score) / 5
    return round(health_score, 2)
```

Figure 203. Pilot 5 Health Index calculation

P5-BP1-DA10 Forecast Engine – Environmental AI Component

As part of Pilot 5's development activities, we designed and integrated a time series forecasting system to predict key environmental conditions in indoor spaces. The forecasting relies on XGBoost regression models, each specialized for a specific variable: temperature, humidity, CO_2, PM1, PM2.5, and PM10. These models use rolling averages and lagged values among other automatically extracted time series features as input from historical environmental sensor data. The feature extractor generates lagged values by shifting past observations (e.g., 1, 2, or 3 steps back) and rolling statistics by applying simple moving averages and other window-based metrics (mean, std, min, max) over the last n timesteps. Both methods treat all values within the window equally, i.e., no additional weight is given to more recent observations. This enables the system to efficiently record transient patterns and variations. The models forecast future values based on room-specific datasets allowing for the early identification of possible declines in air quality. The forecasts that are produced serve as important inputs for downstream components like the optimization engine and room management services in addition to being helpful for display or alerting. Overall, by facilitating the shift from reactive to proactive environmental management this forecasting layer supports smart buildings that are healthier and more adaptable. Below it is shown the list of the trained ML models in the context of Pilot 5, whereas the following figures depict an instance of the ML-based metric prediction for the targeted metrics; namely, temperature, CO2, humidity, PM1, PM2.5, and PM10, respectively.

For the evaluation of the ML model's performance the following metrics have been used:

- **Root Mean Squared Error (RMSE):** Measures the average magnitude of prediction errors, with the same units as the target variable. Lower values indicate better performance. What is considered "good" depends on the scale of the data (e.g., an RMSE of 1°C is excellent for temperature forecasts, but poor if the normal range is only 0–2°C). Based on the results presented below, we can conclude that our models demonstrate strong performance.
- **Coefficient of Determination (R^2):** Indicates how much of the variance in the target variable is explained by the model. Values range from 0 to 1, where 1 means perfect prediction and 0 means no explanatory power (negative values indicate the model is worse than predicting the mean). In practice, $R^2 > 0.7$ is considered strong for most forecasting tasks, though acceptable thresholds vary by domain. Based on the results presented below, we can conclude that our models demonstrate strong performance.

Name	Last commit	Last update
..		
CO2_ppm_pipeline.pkl	initial commit	1 week ago
PM10_ug_m3_pipeline.pkl	initial commit	1 week ago
PM1_ug_m3_pipeline.pkl	initial commit	1 week ago
PM2_5_ug_m3_pipeline.pkl	initial commit	1 week ago
consumption_pipeline.pkl	initial commit	1 week ago
humidity_pipeline.pkl	initial commit	1 week ago
temperature_pipeline.pkl	initial commit	1 week ago

Figure 204. Pilot 5 List of Trained ML Models

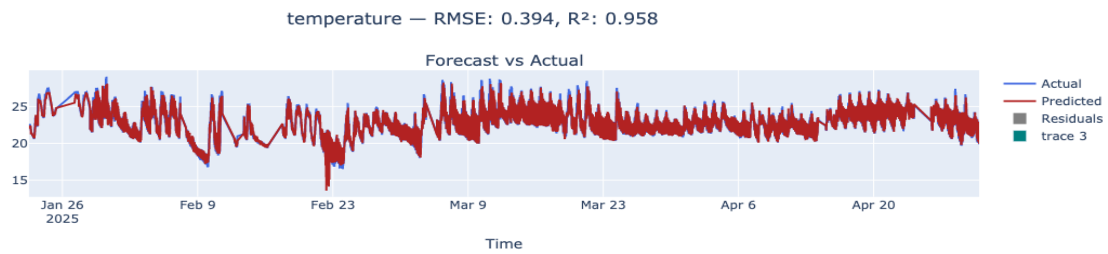


Figure 205. Pilot 5 Evaluation of Temperature Prediction ML Model

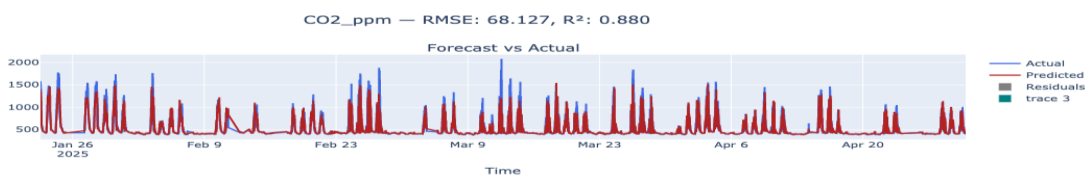


Figure 206. Pilot 5 Evaluation of CO2 Prediction ML Model

Figure 207. Pilot 5 Evaluation of Humidity Prediction ML Model

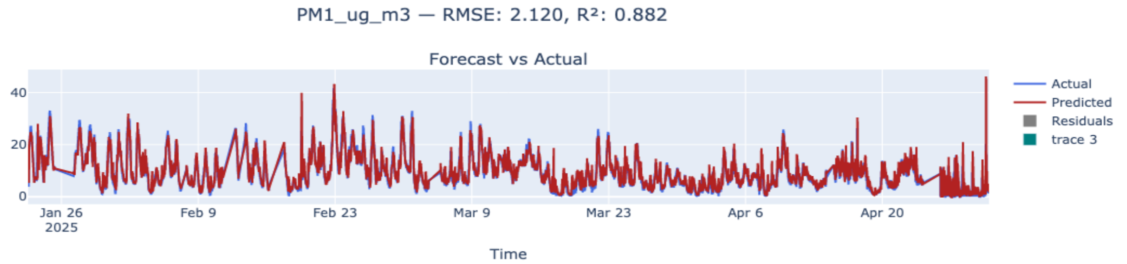


Figure 208. Pilot 5 Evaluation of PM1 Prediction ML Model

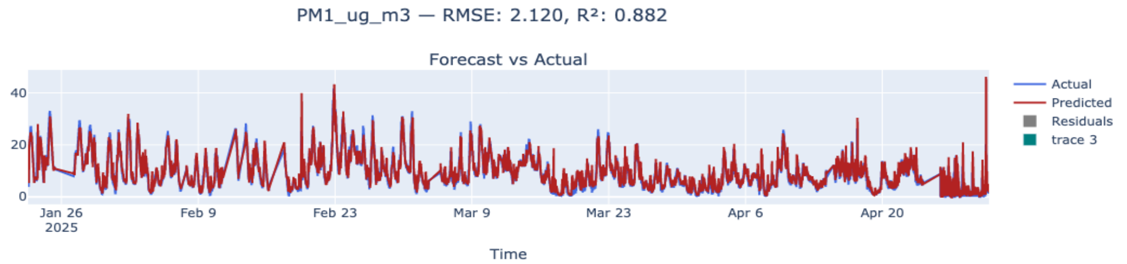


Figure 209. Pilot 5 Evaluation of PM1 Prediction ML Model

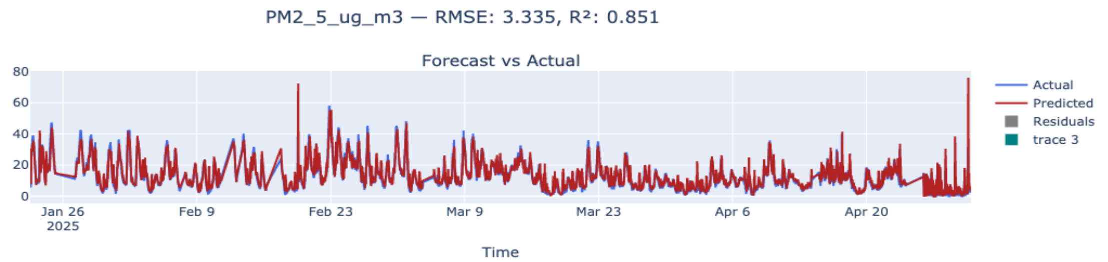


Figure 210. Pilot 5 Evaluation of PM10 Prediction ML Model

P5-BP1-DA11 Forecast Engine - Energy Efficiency AI Component

As part of the development work in Pilot 5, a regression model was implemented to estimate the energy consumption of each room based on environmental and contextual variables. This model uses XGBoost and is trained on a structured dataset that includes features such as temperature, humidity, CO_2 levels, PM, room identity, room volume, and temporal information, such as the hour of the day and the day of the week. Both usage-related and environmental factors that affect energy demand are captured by the model through the encoding of these variables. The output, which is a predicted energy consumption value, can facilitate long-term monitoring reporting and real-time awareness. This regression model, which is applied to individual data samples rather than sequences contrasts with environmental forecasting models and is therefore well-suited for identifying general consumption patterns across rooms and time slots. An instance of the energy-related ML model can be seen in the following figure.

Name	Last commit	Last update
..		
CO2_ppm_pipeline.pkl	initial commit	1 week ago
PM10_ug_m3_pipeline.pkl	initial commit	1 week ago
PM1_ug_m3_pipeline.pkl	initial commit	1 week ago
PM2_5_ug_m3_pipeline.pkl	initial commit	1 week ago
consumption_pipeline.pkl	initial commit	1 week ago
humidity_pipeline.pkl	initial commit	1 week ago
temperature_pipeline.pkl	initial commit	1 week ago

Figure 211. Pilot 5 Energy Consumption Model in the List of ML Models

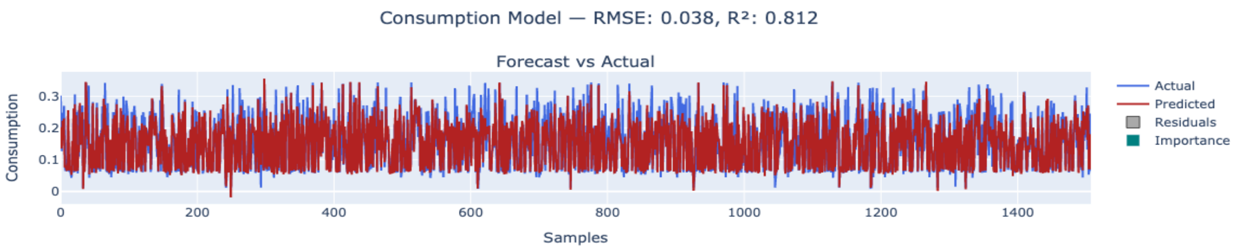


Figure 212. Pilot 5 Evaluation of Energy Consumption Prediction ML Model

P5-BP1-DA12 Health and Energy Optimization

As part of the Pilot 5 development activities, we implemented an optimization component designed to ensure that indoor environmental conditions remain within healthy ranges while minimizing predicted energy consumption. The optimization process takes as input the short-term forecasts of key environmental variables—such as temperature, humidity, CO_2, and PM2.5—alongside energy consumption predictions. It determines whether the predicted conditions differ from established healthy thresholds and if so, calculates the minimal amount of environmental enhancement required to return to ideal conditions at the lowest feasible energy expense. The objective is to define target environmental values that balance energy efficiency and occupant health not to choose particular actions. These optimization results—representing the desired future state of the room—are published to the Orion-LD Context Broker, where they become available to other system components, such as the actuator, which is responsible for interpreting and applying the appropriate control measures. This modular approach supports a clean separation between optimization logic and physical control, while enabling intelligent, goal-driven building automation. The code snippet of the following figure provides the optimizer implementation.

```python
def optimize_environment(room_id: str, room_volume: int):
    day_of_week, hour, prediction_time_str = get_day_and_hour_from_prediction()

    def objective(x):
        temp, humi, co2, pm25, pm10 = x
        data = {
            'day_of_week': day_of_week,
            'hour': hour,
            'room': room_id,
            'volume': room_volume,
            'temperature': temp,
            'humidity': humi,
            'CO2_ppm': co2,
            'PM2_5_ug_m3': pm25,
            'PM10_ug_m3': pm10,
            'time': prediction_time_str
        }

        df = pd.DataFrame([data])[FEATURE_ORDER]
        predictions = model.predict(df)
        value = predictions["consumption_pred"].values[0]
        return value

    x0 = [(low + high) / 2 for low, high in BOUNDS]
    result = minimize(objective, x0, bounds=BOUNDS, method='L-BFGS-B')
    return result, day_of_week, hour, prediction_time_str


def run_optimization(room_id:str, occupancy: int) -> dict:
    """Run optimization for a given room ID and return the results as a dictionary."""

    if room_id.startswith("urn:Pilot5:Room:"):
        room_id = room_id.split(":")[3]
    room_volume = ROOM_VOLUME.get(room_id)

    result, day_of_week, hour, prediction_time_str = optimize_environment(room_id, room_volume)

    if result.success:
        temp, humi, co2, pm25, co = result.x
        predicted_energy = result.fun
        row = {
            'room_id': room_id,
            'datetime': prediction_time_str,
            'room_volume': room_volume,
            'occupancy': occupancy,
            'temp_pred_opt': round(float(temp), 2),
            'humi_pred_opt': round(float(humi), 2),
            'co02_pred_opt': round(float(co2), 2),
            'pm25_pred_opt': round(float(pm25), 2),
            'opt_co': round(float(co), 2),
            'optim_energy_cons': round(float(predicted_energy), 2)
        }

        return row
```

Figure 213. Pilot 5 Health and Energy Optimizer

P5-BP1-DA13 Recommender

The recommender is a software component that generates seat recommendations tailored to the employees' preferences and the energy footprint of the building. The recommender receives as input the building's rooms, the desks of each room along with their availability, the room health score (a metric generated by the AI system), and the employees' preferences (provided by the GUI application). The recommender's output -a list of 3 recommended desks- is provided to the employee via the GUI application. Hence, the recommender sits at the middle point of the Pilot 5 end-to-end system, interacting with both the AI component (indirectly, consuming the health score) and the GUI application (directly via MQTT topics).

The recommendation process is as follows. The recommender always listens on an MQTT topic, that indicates the arrival of an employee, fed by the GUI application. The recommender consumes this information and as soon as an employee arrives, the recommendation process starts. In step 1, all the necessary information as described previously is gathered. In step 2, the recommender creates a list of 3 proposed desks based on the employee’s preferences. This recommendation is rule-based, i.e., a set of 7 rules has been defined, ranging from simple rules, such as Rule 0 “*If the employee's preferred seats are all available, recommend them. The recommendation order is the same with the employee's preference order.*”; to more sophisticated rules, such as Rule 4, which indicates a heuristic to find the most suitable seats based on the employees’ preferences, should some or all of their preferred seats are unavailable. Finally, in step 3, the recommendation list is re-arranged by considering the health score of the room, where each recommended desk is located. A 5-level health score scale is created by setting the minimum as the minimum health score of all the rooms, the maximum as the maximum health score of all the rooms and setting equidistant levels in-between. The 3 recommended desks from step 2 are rearranged in step 3 according to this scale, targeting to favor the desks that are in rooms with better health scores.

When the recommendation process is finalised, the recommender pushes this information to the MQTT server and informs the proper MQTT topic for the completion of the recommendation. Both outputs will be consumed by the GUI application; the recommendation will be depicted to the employee, whereas the notification on the completion of the process will be used as a flag by the GUI application to retrieve the recommended desks.

The recommender component had been finalized during the Technical Meeting on Feb. 2025 in Athens, and since then undergoes testing (see Figure), improvements (see Figure), and has been integrated to the entire Pilot 5 end-to-end system. It has been developed using the Python programming language, with little needs of external libraries, all of which are documented in a requirements file for easy installation inside a virtual environment.

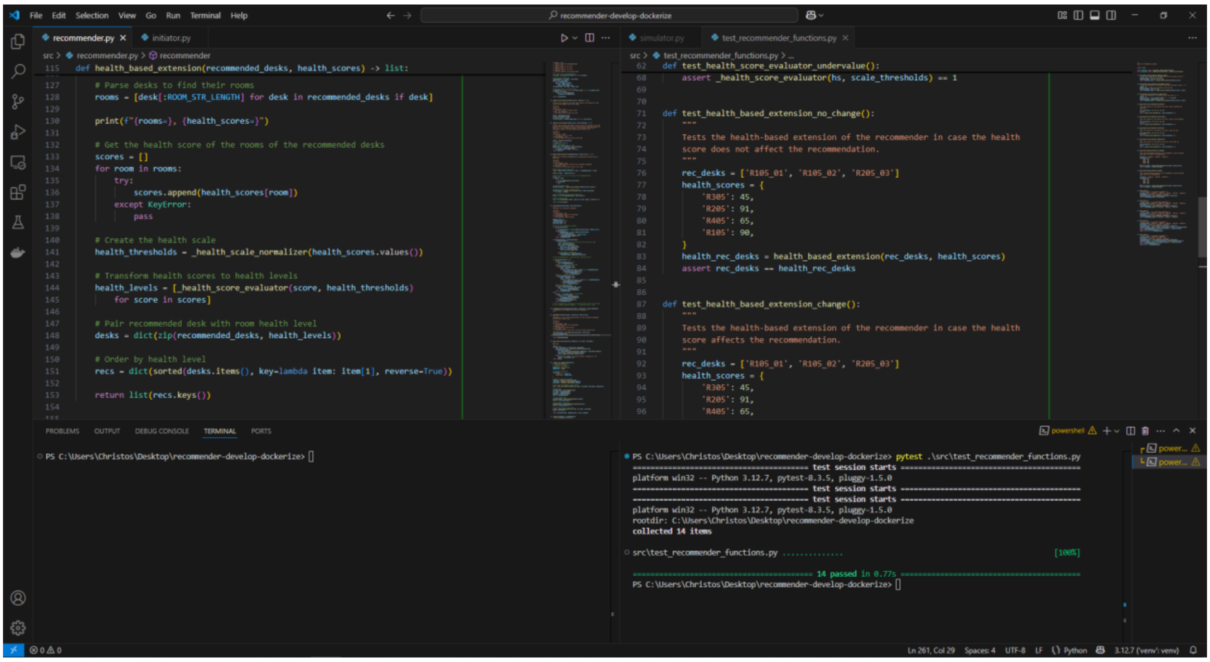


Figure 214. Pilot 5 Recommender testing using pystest library

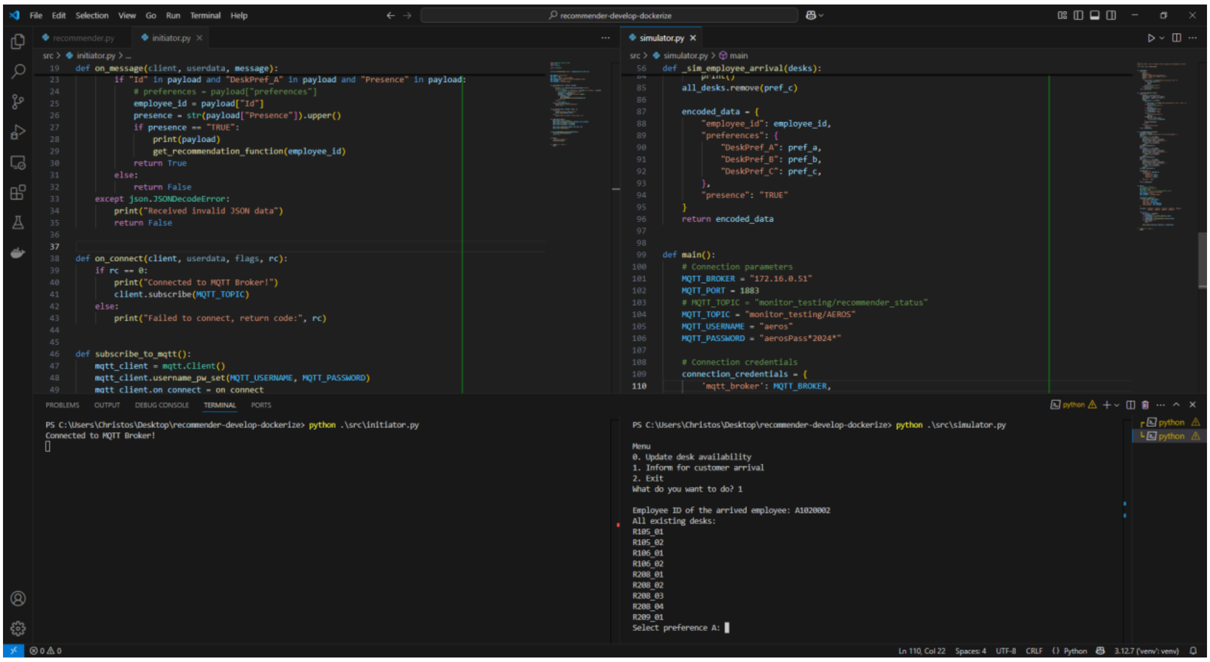


Figure 215. Pilot 5 Recommender simulation to improve its functionality

The recommender acts as a modular component having been containerized using Docker (see Figure) and packaged using Helm. The component has been installed successfully in the OTE infrastructure, which hosts the Pilot 5 end-to-end system.

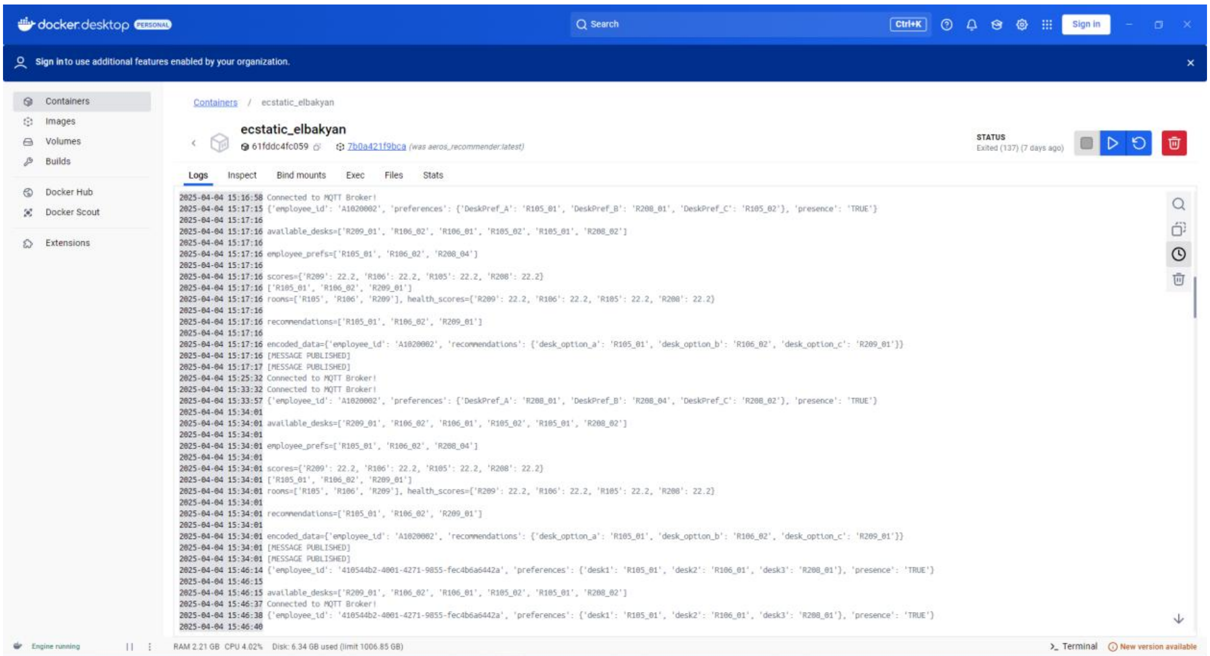


Figure 216. Pilot 5 Recommender container built and started as seen in Docker container

P5-BP1-DA14 End-user GUI Application

The end-user graphical interface has been developed to support authenticated users' interaction with the smart building services of Pilot5. The GUI offers key functionalities, such as personalized desk recommendations, workspace booking, and preference management. The interface is responsive and designed for seamless operation across various devices.

Figure 217. Pilot 5 Home Page

User interactions, such as presence updates or desk preferences, are stored in a local MySQL database, which is integrated with the aerOS Data Fabric. After detecting updates (e.g., presence status set to *True*), the Recommender system triggers a processing flow and publishes a new desk recommendation to the Orion-LD context broker for the dedicated user, named by his unique ID number. The GUI fetches the relevant recommendation entities via standard NGSI-LD queries and renders them dynamically to the user.

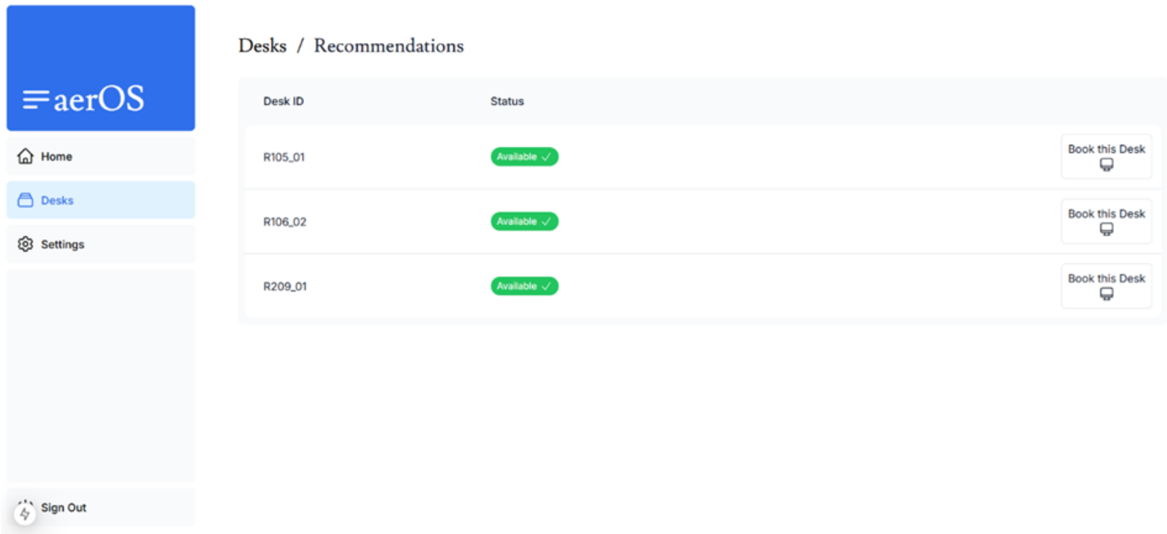


Figure 218. Pilot 5 Desk Recommendation Page

The interface reflects real-time data consistency and has been validated in live pilot conditions. It is continuously enhanced based on participant user's feedback, with a focus on usability and contextual awareness.

2.5.1.1.3. Integration activities

P5-BP1-IA15 Integration of aerOS Basic IE Components

The Basic IE Components installed in Pilot 5 are listed in section 3 aerOS components assessment. Figure depicts the relevant running processes.

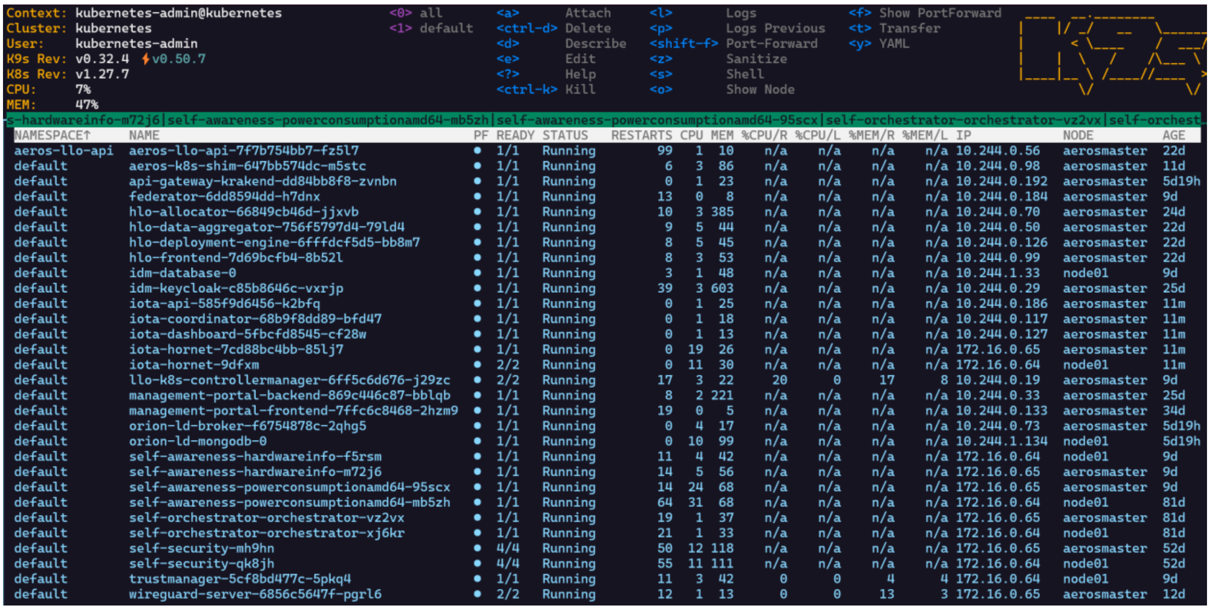


Figure 219. Pilot 5 aerOS Basic Components running in Pilot 5 Domain 01: Entry Domain

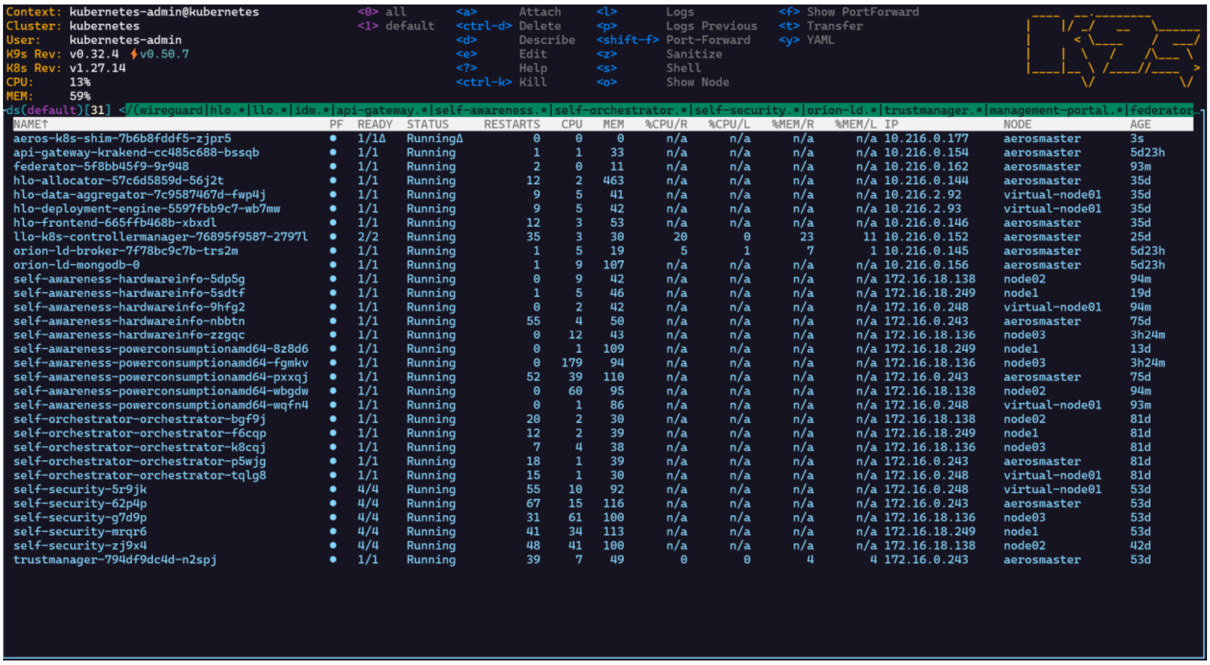


Figure 220. Pilot 5 aerOS Basic Components running in Pilot 5 Domain 02: Main Domain

As shown in Figure , the aerOS Entry domain offers three means for the users to interact with the system involving the respective basic components.

While Kraken-D offers an API for configuration management, the aerOS portal, the Keycloak and Open LDAP expose a portal for the users' interaction.

These portals are appropriately installed in the Pilot5 domain as depicted in Figure , Figure and Figure respectively.

Figure 221. Pilot 5 aerOS Management Portal

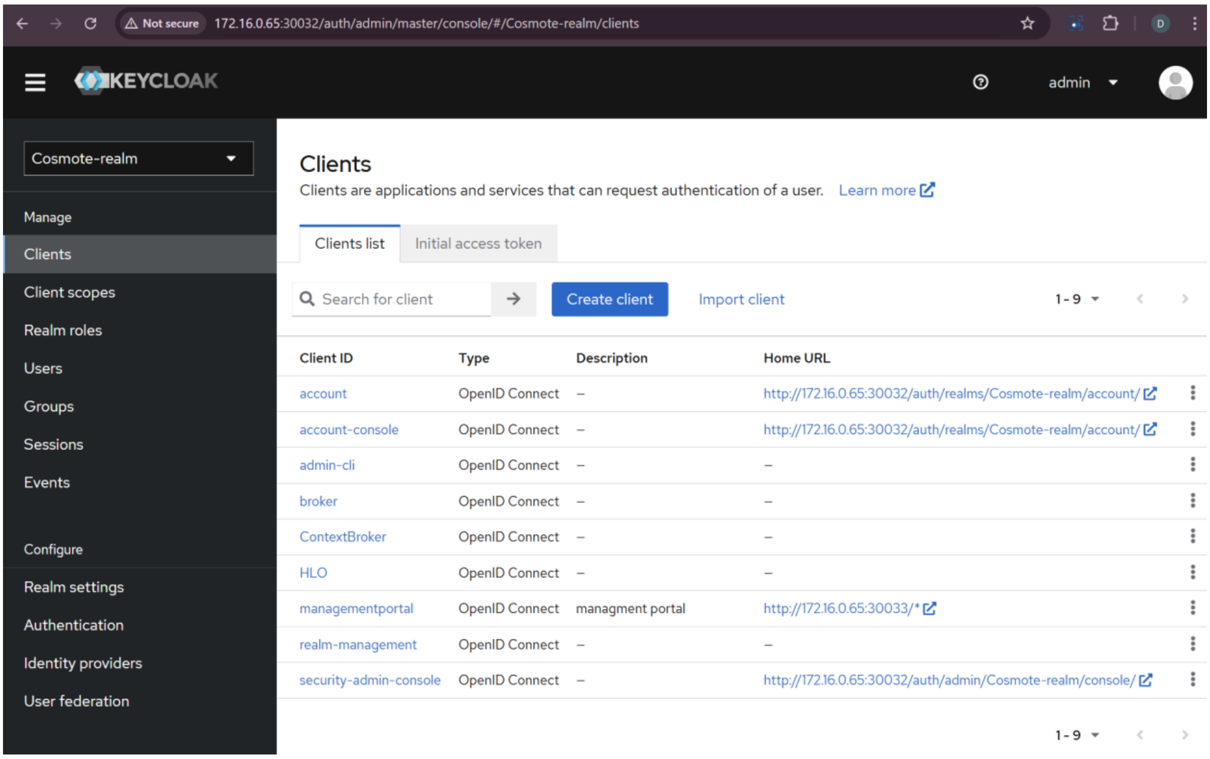


Figure 222. Pilot5 aerOS Keycloak Portal

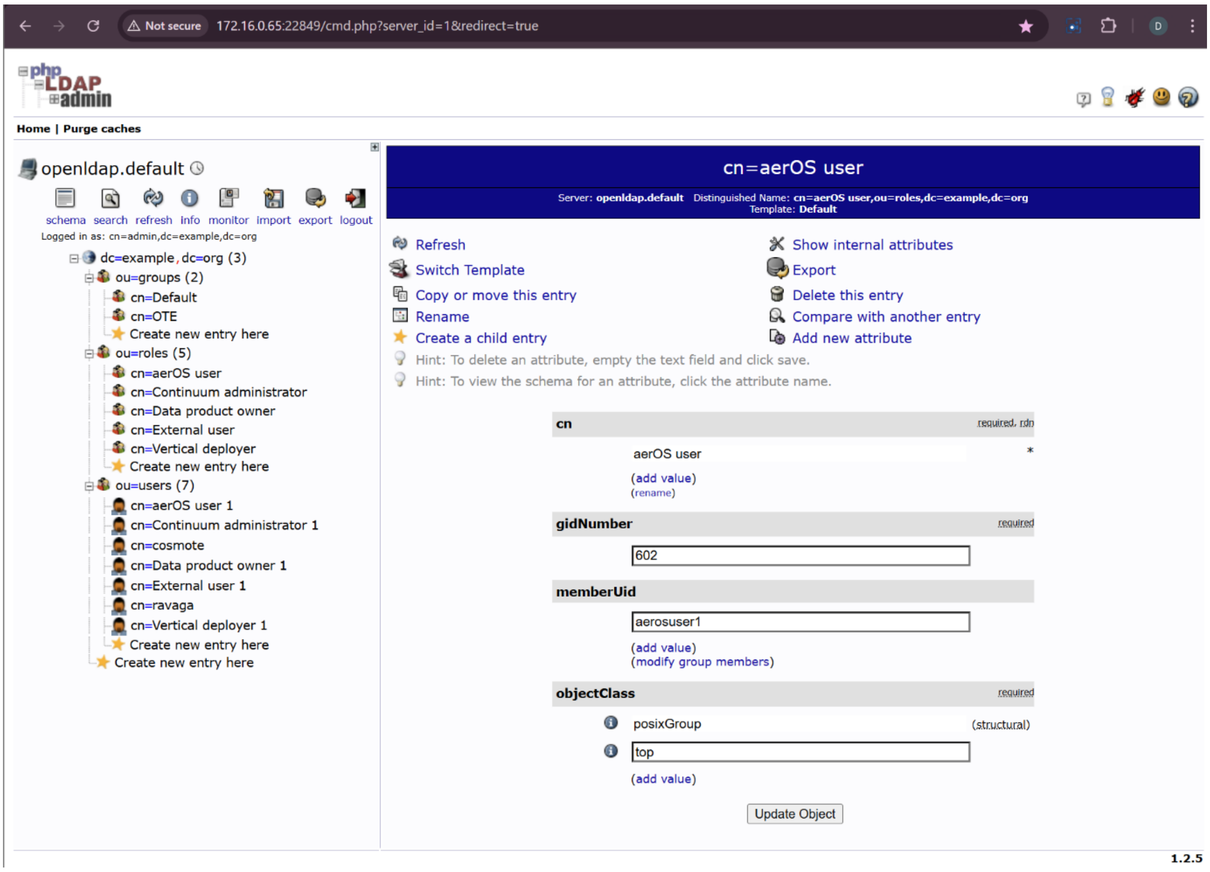


Figure 223. Pilot5 OpenLDAP Portal

P5-BP1-IA16 Integration of aerOS non-Basic IE Components

The non-Basic IE Components installed in Pilot 5 are listed in section 3 aerOS Component Assessment. Figure and Figure depict the relevant running processes per respective Pilot 5 domain.

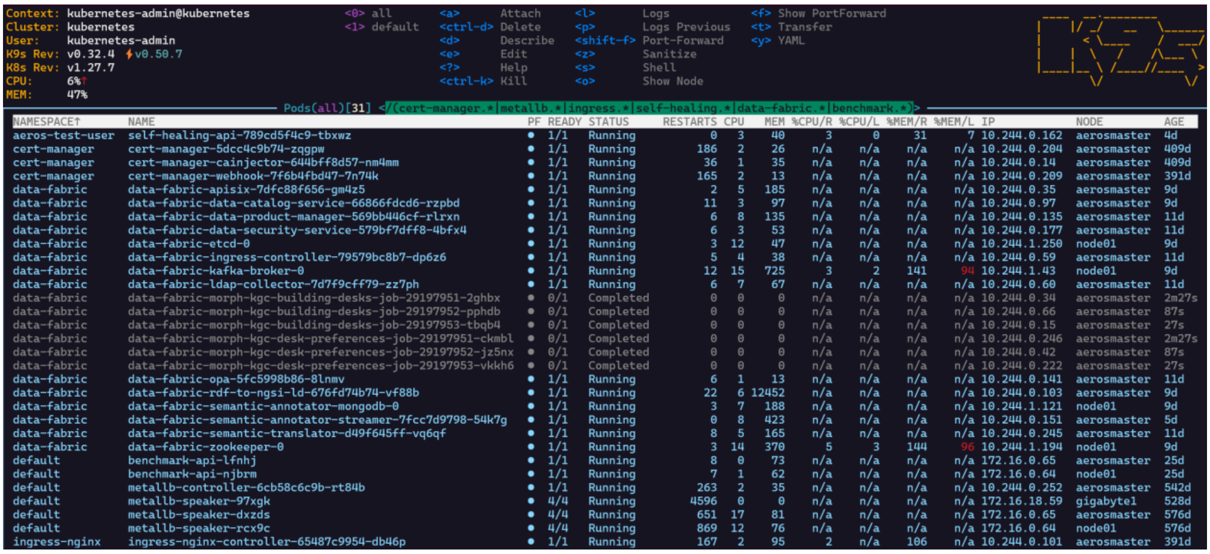


Figure 224. Pilot 5 aerOS non-Basic Components running in Pilot 5 Domain 01: Application Domain

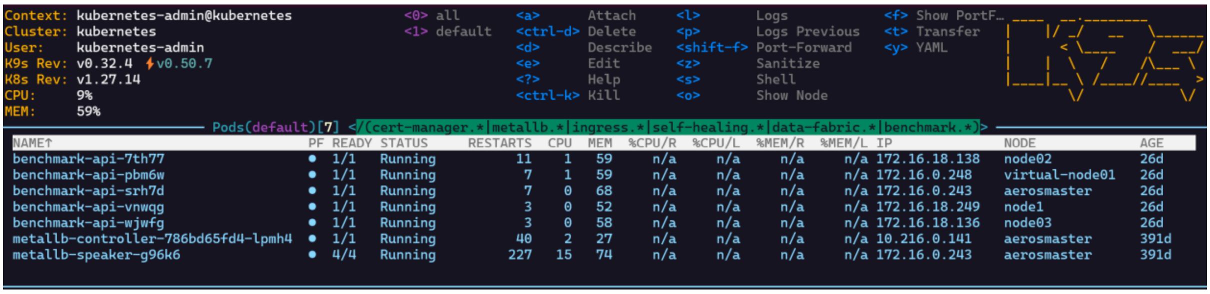


Figure 225. Pilot 5 aerOS non-Basic Components running in Pilot 5 Domain 02: Main Domain

It must be noted that the data fabric developments of the project, while considered non-Basic components have a significant role in the message exchanges between the Pilot 5 application components. Figure shows the configuration of the Orion-LD with all the Pilot application data products.


```json
[
    {
        "id": "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:Catalog",
        "type": "http://www.w3.org/ns/dcat#Catalog",
        "https://w3id.org/aerOS/continuum#domain": {
            "type": "Relationship",
            "object": "urn:ngsi-ld:Domain:Domain01"
        },
        "http://www.w3.org/ns/dcat#themeTaxonomy": {
            "type": "Relationship",
            "object": "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:BusinessGlossary"
        },
        "https://w3id.org/aerOS/data-catalog#contextBroker": {
            "type": "Relationship",
            "object": "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:ContextBroker"
        },
        "https://w3id.org/aerOS/data-catalog#dataProduct": {
            "type": "Relationship",
            "object": [
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:2a99d45e-3c21-44a0-891c-e602c4af7609",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:c478486d-7bdf-451a-9580-0013a194bb60",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:4c0aedb5-196f-45e8-95c1-681e6f36860f",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:4369979a-8cbf-4847-bdb8-729b47ec53ab",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:8eb05027-92b8-400e-bb32-984110d1a3eb",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:6fc0b1b7-2858-446a-ae8a-539b80a182b1",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:4a147e32-a6ac-4e13-a6bc-bab692956279",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:105f5b18-aa31-4de6-9f17-39d67f5d7913",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:b53df142-199e-41fd-8cc8-760a351d0db3",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:3478c0c8-d9dc-4bdf-8f8c-fbf9b2ad9675",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:f034c76a-268b-41e2-83cf-4a6e431ccdbf",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:f19d97e5-3da2-4851-9ccc-85ab1247651f",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:7788183f-013a-4a77-b1e0-19a7ea3fdb95",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:e2be9fbd-2d6b-4708-9320-5cf06301925c",
                "urn:ngsi-ld:Organization:Cosmote-dev:Domain:Domain01:DataProduct:c5935623-beb0-4f63-9e57-49d068588061"
            ]
        }
    }
]
```


Figure 226. Pilot 5 Orion LD Pilot5 data products

P5-BP1-IA17 Integration of Data Fabric with the IoT backend

Regarding the IoT data, within the aerOS data fabric and the OrionLD system, 11 data products have been defined and tested, as shown in Figure below.

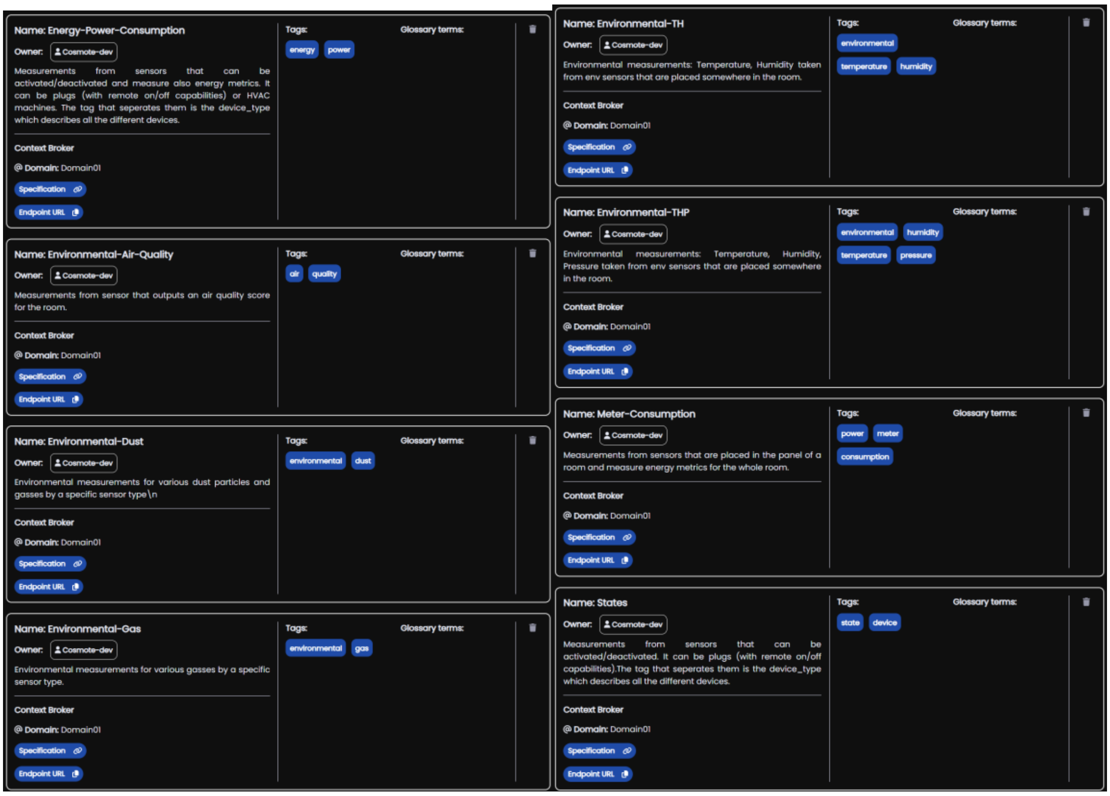


Figure 227. Pilot 5 IoT Data Products in aerOS Data Fabric (Catalog)

P5-BP1-IA18 Integration of Data fabric with Forecast Health Index System

The Forecast Health Index component was successfully integrated with the aerOS Data Fabric. The computed health scores are continuously published to the Orion-LD Context Broker, making them available to downstream components, such as the Recommender and GUI.

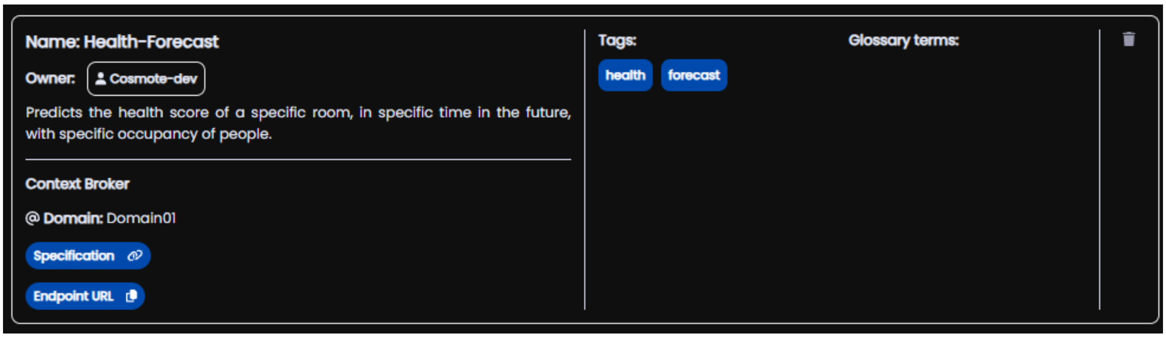


Figure 228. Pilot 5 Forecast Health Index integrated with aerOS Data Fabric

P5-BP1-IA19 Integration of Data fabric with Forecast Environmental AI System

The Forecast Environmental AI component has been integrated with the aerOS Data Fabric, enabling continuous publication of predicted environmental values (e.g., temperature, humidity, CO_2, PM) to the Orion-LD Context Broker. These forecasts are accessible to other components for real-time awareness and proactive decision-making.

Figure 229. Pilot 5 Forecast Environmental AI integrated with aerOS Data Fabric

P5-BP1-IA20 Integration of Data Fabric with Forecast Energy Efficiency System

The Forecast Energy Efficiency component has been integrated with the aerOS Data Fabric, allowing predicted room-level energy consumption to be published to the Orion-LD Context Broker. This information supports downstream components such as the optimization system in making energy-aware decisions.

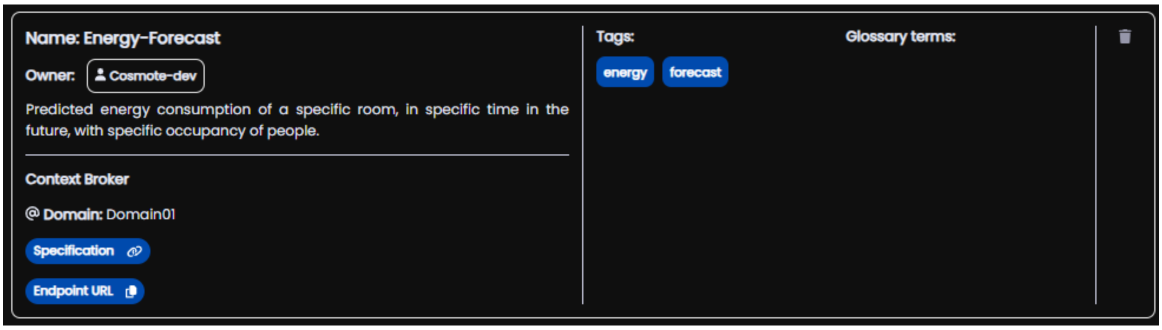


Figure 230. Pilot 5 Forecast Energy Efficiency System integrated with aerOS Data Fabric

P5-BP1-IA21 Integration of Data fabric with Optimization System

The Optimization component has been integrated with the aerOS Data Fabric to publish computed target environmental conditions for each room. These optimization results are made available through the Orion-LD Context Broker and are consumed by the actuator system for execution.

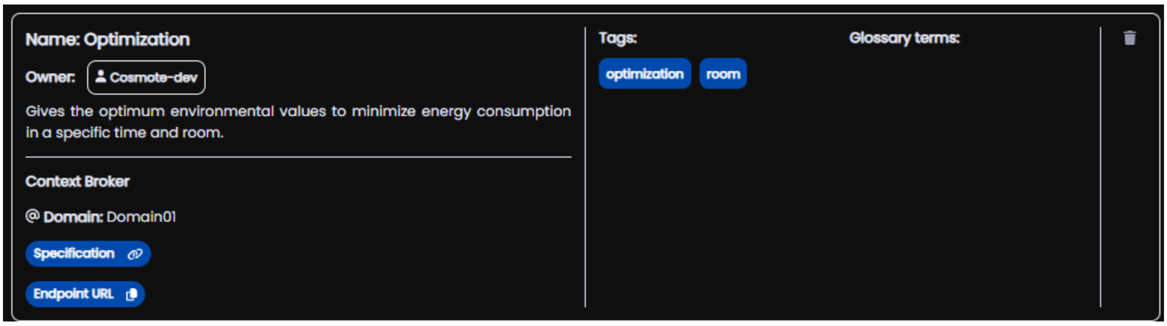


Figure 231. Pilot 5 Optimization System integrated with aerOS Data Fabric

P5-BP1-IA22 Integration of Data Fabric with Recommender System

Recommender has been tested with and integrated in the Data Fabric. The required information for connection purposes is stored in external files and provided in the recommender thanks to the Helm packaging, whereas the required information for the recommendation process is parsed properly.

For connection purposes the recommender connects to three MQTT topics regarding (i) the arrival of a new employee, (ii) the consumption of the necessary information for the recommendation (rooms, desks, health scores, etc.) and the production of the recommended seats, and (iii) the notification of the GUI. This consumed and produced information is stored in the Orion-LD. The connection credentials are in the values.yaml file of the Helm packaging.

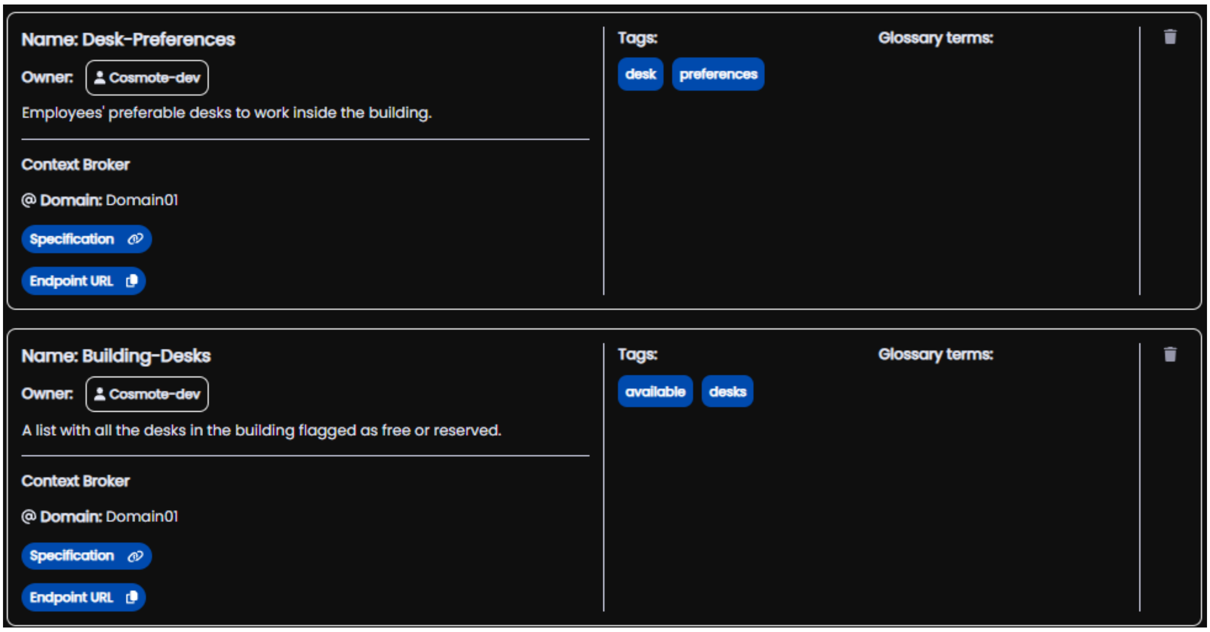


Figure 232. Pilot 5 Recommender consumes the employee's preferences and the desks of the building from the Data Fabric

For the parsing of the information, appropriate modules have been developed to (i) fetch and (ii) decode the necessary information. The modules haven been developed having in mind the ontology of Pilot 5. Furthermore, the reverse operations of encoding and sending have also been completed, since the recommender apart from consuming data, also produces the output of the recommendation.

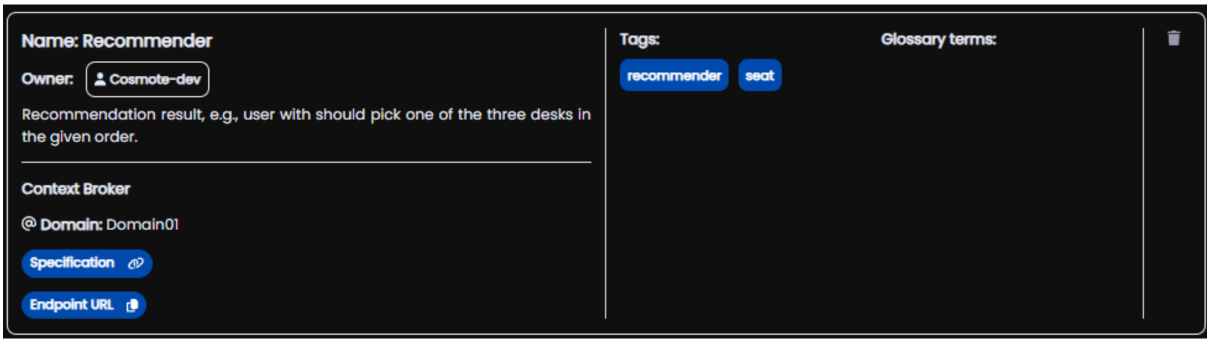


Figure 233. Pilot 5 Recommender produces the recommended desks to the Data Fabric

P5-BP1-IA23 Integration of Data Fabric with End-user GUI System

The end-user GUI system of Pilot 5 has been successfully integrated with the aerOS Data Fabric, enabling seamless interaction between the user interface and data infrastructure. The GUI plays a dual role: (i) submitting user inputs that get inserted into the Data Fabric, and (ii) querying real-time data from Orion-LD (provided by Recommendation system) to present updated information to the user.

Storing user input into the Data Fabric

When a user interacts with the GUI -by updating presence or desk preferences- these changes are stored in a local MySQL database on OTE's cluster. The following figure shows how the user presence is updated through a dedicated API route.

```
import mysql from 'mysql2/promise';
import { NextResponse } from 'next/server';
import { auth } from '@/auth';

const connectionPool = mysql.createPool({
  host: '172.16.0.96',
  user: 'db_user',
  password: 'userPass!',
  database: 'app_db',
  waitForConnections: true,
  connectionLimit: 10,
  queueLimit: 0,
});

export async function POST() {
  const session = await auth();
  const userId = session?.user?.id;

  if (!userId) {
    return NextResponse.json({ error: 'User ID not found' }, { status: 401 });
  }

  const conn = await connectionPool.getConnection();
  try {
    await conn.query(
      'UPDATE EMPLOYEES_PREFERENCES SET Presence = true WHERE Id = ?',
      [userId]
    );
    conn.release();
    return NextResponse.json({ message: 'Presence updated to true' });
  } catch (error) {
    conn.release();
    console.error('Error updating presence:', error);
    return NextResponse.json({ error: 'Failed to update presence' }, { status: 500 });
  }
}
```

Figure 234. Pilot 5 Set-Presence-True API

Similarly, desk preference updates are handled and persisted using the following logic. These updates are synchronized with the aerOS Data Fabric, which detects changes and pushes them into the Orion-LD context broker in the form of NGSI-LD entities. In this way, Recommender is getting notified and can start the recommendation process.

Retrieving data from Orion-LD via NGSI-LD queries

The GUI system also retrieves recommendation results and other contextual data directly from Orion-LD. Upon receiving a user request, the application formulates dynamic NGSI-LD queries to fetch the appropriate entities. This is shown in the figure of the recommendation API route.

```
import { NextResponse } from "next/server";

export async function GET(request: Request) {

  const { searchParams } = new URL(request.url);
  const userId = searchParams.get("user");

  if (!userId) {
    return NextResponse.json({ error: "Missing user ID" }, { status: 400 });
  }

  const baseUrl = "http://172.16.0.65:13646/ngsi-ld/v1/entities";
  const slots = [1, 2, 3];
  const employeeId = `urn:Pilot5:Employee:${userId}:RankedRecommendation:Slot`;

  try {
    const fetchPromises = slots.map(async (desk) => {
      const url = `${baseUrl}/${employeeId}:${desk}`;
      console.log(`Fetching: ${url}`);

      const response = await fetch(url, {
        method: "GET",
        headers: {
          Accept: "application/json",
        },
      });

      console.log(`Response for slot ${desk}: ${response.status}`);

      if (!response.ok) {
        throw new Error(`HTTP error! Status: ${response.status} for slot ${desk}`);
      }

      const data = await response.json();

      const objectValue = data["http://www.w3.org/ns/org#item"].object;
      const extractedString = objectValue.split(":").pop();

      return { desk, object: extractedString };
    });

    const results = await Promise.all(fetchPromises);

    console.log("Final recommendations:", results);

    return NextResponse.json({ recommendations: results });
  } catch (error) {
    console.error("Error fetching recommendations:", error);
    return NextResponse.json(
      { error: "Failed to fetch recommendations" },
      { status: 500 }
    );
  }
}
```

Figure 235. Get-Recommendation API

The retrieved recommendation slots are then displayed to the user through the GUI in real-time. This interaction pattern ensures that the application remains fully aligned with the state maintained by the aerOS Data Fabric and leverages the shared semantic model provided by Orion-LD.

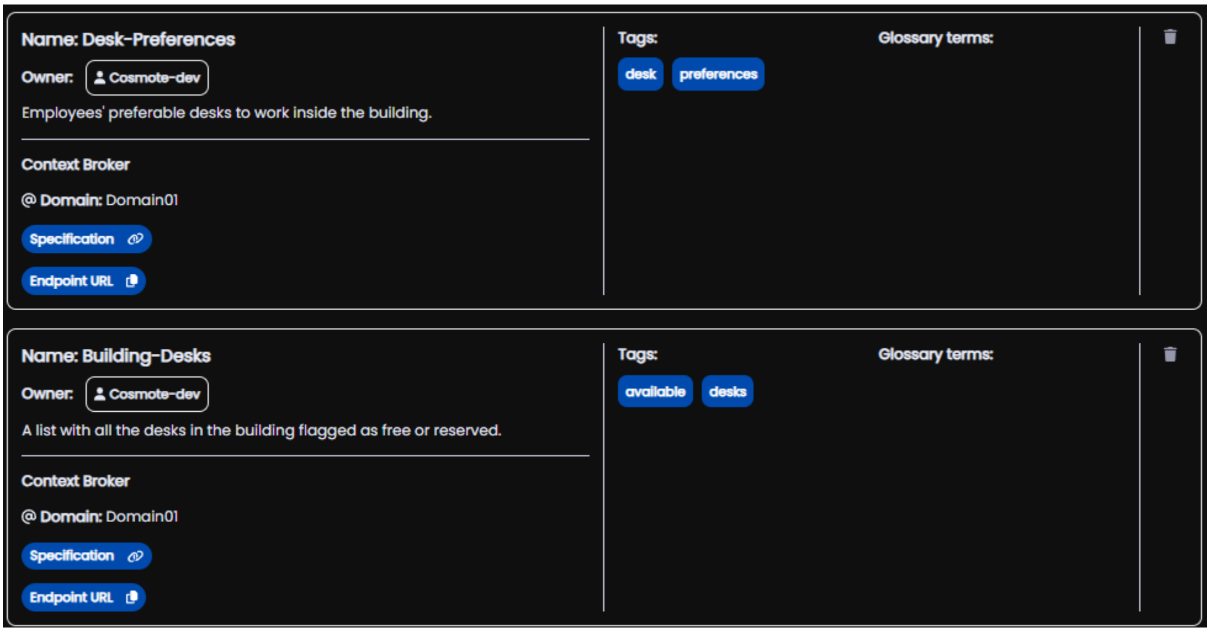


Figure 236. Pilot 5 Data Products - GUI

P5-BP1-IA24 E2E Integration of all Application Components (IoT, Forecast Engine, Recommender, GUI)

The end-to-end integration can be demonstrated through the execution of a full end-user cycle involving the end-user approaching to the building (logging in and authenticated in the Web GUI) and then proceeding to receive the seating recommendations. This cycle implies the appropriate integration among all components so that the recommendation is produced and presented to the user, as can be seen through the respective log file of the GUI and depicted in Figure .

What is not clearly visible from these logs and is depicted in Figure , is that the seating recommendations are produced using the Health Index calculations as produced by the Forecasting AI component of the pilot.

```
Database user: {
  id: '312532b2-4001-4271-9855-fec4b6a6442a',
  name: 'Dimitris Salmatanis',
  email: 'dsalmatanis@oteresearch.gr',
  password: '$2b$10$ug2r4OGtgU6hL/CkAN3I7uXVuuSiWFP8HzqWozkLwqv9k7tM3.nUm'
}
Authorized user: {
  id: '312532b2-4001-4271-9855-fec4b6a6442a',
  name: 'Dimitris Salmatanis',
  email: 'dsalmatanis@oteresearch.gr'
}
MQTT SEND → Running: python "/app/scripts/send_to_mqtt.py" "{\"Id\":\"312532b2-4001-4271-9855-fec4b6a6442a\",\"DeskPref_A\":\"R105_02\",\"De
skPref_B\":\"R106_01\",\"DeskPref_C\":\"R209_01\",\"Presence\":1,\"Rec_System_Rating\":0}"
MQTT SEND → Success:

API /api/mqtt called at 2025-07-09T11:07:28.034Z
Child process started: 62
data
data
GET /API/RECOMMENDATION REST CALL
➡️ Fetching: http://172.16.0.65:13646/ngsi-ld/v1/entities/urn:Pilot5:Employee:312532b2-4001-4271-9855-fec4b6a6442a:RankedRecommendation:Slot
:1
➡️ Fetching: http://172.16.0.65:13646/ngsi-ld/v1/entities/urn:Pilot5:Employee:312532b2-4001-4271-9855-fec4b6a6442a:RankedRecommendation:Slot
:2
➡️ Fetching: http://172.16.0.65:13646/ngsi-ld/v1/entities/urn:Pilot5:Employee:312532b2-4001-4271-9855-fec4b6a6442a:RankedRecommendation:Slot
:3
⟵ Response for slot 1: 200
📦 Data for slot 1: {
  "id": "urn:Pilot5:Employee:312532b2-4001-4271-9855-fec4b6a6442a:RankedRecommendation:Slot:1",
  "type": "http://www.w3.org/ns/org#Slot",
  "http://www.w3.org/ns/org#index": {
    "type": "Property",
    "value": "1"
  },
  "http://www.w3.org/ns/org#item": {
    "type": "Relationship",
    "object": "urn:Pilot5:Desk:R209_01"
  }
}
⟵ Response for slot 3: 200
📦 Data for slot 3: {
  "id": "urn:Pilot5:Employee:312532b2-4001-4271-9855-fec4b6a6442a:RankedRecommendation:Slot:3",
  "type": "http://www.w3.org/ns/org#Slot",
  "http://www.w3.org/ns/org#item": {
    "type": "Relationship",
    "object": "urn:Pilot5:Desk:R106_01"
  },
  "http://www.w3.org/ns/org#index": {
    "type": "Property",
    "value": "3"
  }
}
⟵ Response for slot 2: 200
📦 Data for slot 2: {
  "id": "urn:Pilot5:Employee:312532b2-4001-4271-9855-fec4b6a6442a:RankedRecommendation:Slot:2",
  "type": "http://www.w3.org/ns/org#Slot",
  "http://www.w3.org/ns/org#index": {
    "type": "Property",
    "value": "2"
  },
  "http://www.w3.org/ns/org#item": {
    "type": "Relationship",
    "object": "urn:Pilot5:Desk:R105_02"
  }
}
✅ Final recommendations: [
  { desk: 1, object: 'R209_01' },
  { desk: 2, object: 'R105_02' },
  { desk: 3, object: 'R106_01' }
]
```

Figure 237. Pilot 5 End-to-end user interaction – Web GUI logs

```
{'metadata': {'Content-Type': 'application/json', 'User-Agent': 'orionld/post-v1.9.0', 'Host': 'aeros:0', 'Accept': 'appl
ication/json', 'Ngsild-Attribute-Format': 'Normalized', 'Link': 'https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.
8.jsonld'}, 'body': {'id': 'urn:ngsi-ld:Notification:fc346156-5cb4-11f0-a936-2698f97b43c2', 'type': 'Notification', 'subs
criptionId': 'urn:aeros:subscription:presence:pilot5', 'notifiedAt': '2025-07-09T11:08:04.359Z', 'data': [{'id': 'urn:Pil
ot5:Employee:312532b2-4001-4271-9855-fec4b6a6442a', 'type': 'https://w3id.org/aerOS/building#Employee', 'https://w3id.org
/aerOS/building#presentInBuilding': {'type': 'Property', 'value': 'true'}}]}}

available_desks=['R106_02', 'R106_01', 'R208_03', 'R208_01', 'R105_01', 'R209_01', 'R208_04', 'R105_02', 'R208_02']

employee_prefs=['R105_02', 'R106_01', 'R209_01']

scores={'R209': 80.0, 'R105': 79.91000366210938, 'R106': 79.75, 'R208': 79.9800033569336}
rule_based_desks=['R105_02', 'R106_01', 'R209_01']
rooms=['R105', 'R106', 'R209'], health_scores={'R209': 80.0, 'R105': 79.91000366210938, 'R106': 79.75, 'R208': 79.9800033
569336}

recommendations=['R209_01', 'R105_02', 'R106_01']
[MESSAGE PUBLISHED]
[MESSAGE PUBLISHED]
```

Figure 238. Pilot 5 Forecasting and Recommender Interaction - Recommender logs

2.5.2. Cybersecurity and data privacy in building automation

Scenario 2 of **Pilot 5** focuses on the definition and implementation of access control mechanisms within the **aerOS MetaOS** framework, emphasizing how these mechanisms can be adapted to secure the system's functionality across **OTE's** domains. This scenario explores the secure orchestration and management of IoT systems and services, ensuring that various stakeholders can interact with the system based on their specific responsibilities. At the same time, it explores the capabilities of **aerOS** as a **MetaOS** to support the extension of **5G cores**, enabling network functions to be managed across the continuum. This includes support for the deployment of network functions at the edge, such as **UPF**, and the management of network functions that can securely interact with the **5G core** providing operational data that can support new applications development. It seeks to demonstrate how **aerOS MetaOS** can provide a flexible, secure environment for managing both network and IoT functions, while facilitating new application development based on insights from 5G-enabled environments.

2.5.2.1. Technical Schema

As described in Scenario 1 above, Pilot 5 operates across two aerOS domains. One domain, the **Entrypoint aerOS Domain**, hosts 2 IEs (Integrated Elements) and is responsible for hosting the applications. The second domain, the **Edge Domain**, has 4 integrated IEs and supports **KubeEdge**, which enables the integration, access, and management of IoT devices (sensors and actuators). This domain facilitates both metrics collection and actuations control.

While the two domains have separate functional roles in the use case and Scenario 1 deployment, they share the same set of API endpoints for capabilities exposure and resource access, following the aerOS architecture. In addition to these shared capabilities, the Entry point domain integrates **AAA (Authentication, Authorization, and Accounting)** functionalities, as described in **D2.6** and **D2.7**. These AAA functionalities enable and enforce cyber-security policies based on roles and resources.

An abstract representation of this topology is presented in below, illustrating the two domains, their integrated IEs, and the exposed APIs. The figure clearly depicts the layered interactions between these components.

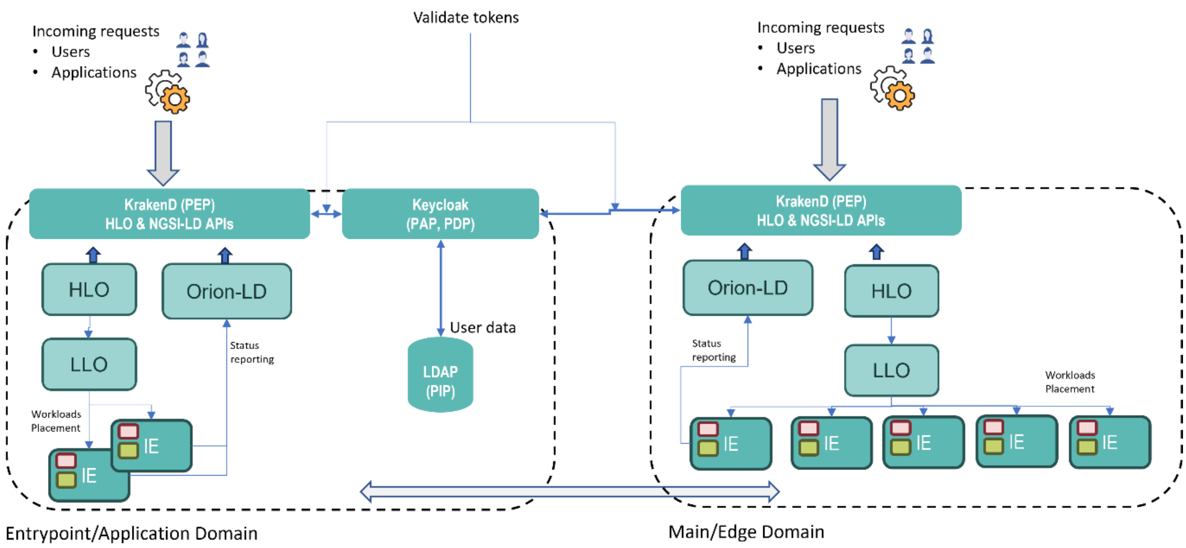


Figure 239. Pilot5 Cyber-Security Components

As discussed in the WP2, WP3, and WP4 deliverables, all aerOS APIs are REST-based. The **HLO APIs** can be used to manage and orchestrate (LCM) services across the continuum, while the **NGSI-LD APIs** allow for retrieving, updating, or deleting information regarding both IEs' availability and capabilities, as well as (edge) IoT data.

Given the increased interconnectivity of IoT systems deployed on top of aerOS IEs, which enable automation, optimization, and monitoring of building functions such as HVAC, lighting, and security, this expanded connectivity also introduces potential security risks and vulnerabilities. This second scenario has been selected to validate that aerOS provides a flexible **cybersecurity suite** that offers granular control over security. This suite can manage and control access to both computing resources and continuum data, addressing the security concerns introduced by the growing connectivity of IoT systems.

The components used to integrate cybersecurity functionality are illustrated in Figure and align with the standardized system functionalities of **PIP**, **PAM**, **PDP**, and **PEP**, each of which has the following roles:

- **PIP (Policy Information Point)**: In aerOS, **LDAP** serves as the PIP, providing user identity data, roles, and attributes that are used by the PDP to make policy decisions.
- **PAM (Policy Administration Point)**: **Keycloak** acts as the PAM in aerOS, managing and administrating security policies, including defining roles, permissions, and authentication rules.
- **PDP (Policy Decision Point)**: In aerOS, **Keycloak** also functions as the PDP, evaluating policies and making access decisions based on the data provided by the PIP (LDAP) and the policies created in the PAM.
- **PEP (Policy Enforcement Point)**: **Krakend** serves as the PEP in aerOS, enforcing the access control decisions made by the PDP and ensuring that only authorized actions are allowed on the system's resources

While the above scenario is used to validate security and trust within the aerOS domain, another aspect of this scenario focuses on evaluating the secure extension of 5G network capabilities across the aerOS continuum. As OTE is an LTE/5G service provider, it was a challenge to validate aerOS as a programmable environment capable of hosting VNFs closely related to 5G services. In the figure below, the designed integration is illustrated, which supports the on-demand deployment of VNFs. This integration ensures secure interactions with external 5G cores—**in this case, Open5gs** was chosen for the evaluation.

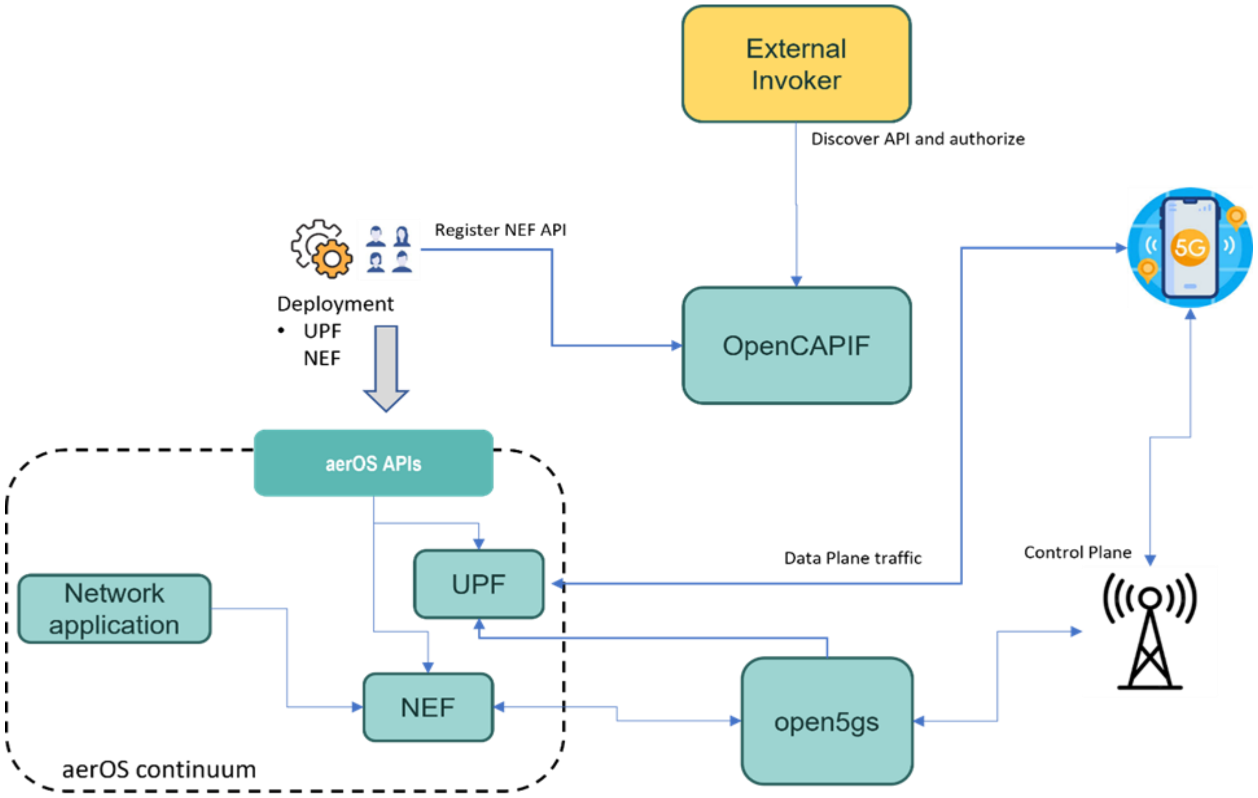


Figure 240. Extending 5G over aerOS

Furthermore, as illustrated in the figure above, to enhance the security and interoperability of the 5G services deployed within the aerOS continuum, we also deployed an OpenCAPIF instance, which implements the 3GPP Common Application Programming Interface Framework (CAPIF) standard. The OpenCAPIF has a dual role, acting in one hand as a registry to discover deployed VNF services, and providing a standardized and secure interface for exposing these VNFs to external invokers.

2.5.2.1. Report of activities

2.5.2.1.1. Setup and procurement activities

P5-BP2-SA1 Procurement of Server and Equipment

This scenario is based on the overall computing provisions that are also required, and described, in scenario 1, and have been secured and / or procured. Primarily this relates with availability of computing resources for the deployment of aerOS continuum (already addressed in scenario 1). No extra equipment is needed for this scenario.

P5-BP2-SA2 Identification of targeted 5G core.

As a network provider, OTE has a strong interest in a multitude of 5G cores both open sourced and commercial. While functionalities exposed and concepts might be common (as described in 3GPP specifications), each core usually supports a custom interface to interface with its capabilities. For the implementation of our VNFs, open5gs was selected, as it is a widely used implementation and can perform both as a single instance and as a multi containers deployment. The selection of open5gs core as a targeted core provided the insights of the VNFs that could be to interact with it.

2.5.2.1.2. Development activities

P5-BP2-DA1 Installation of aerOS Domains

This is a common activity for both scenarios, as the existence of aerOS continuum is also a prerequisite for this, second, scenario. So, as also reported in the previous one, aerOS domains that are part of the pilot architecture are include Domain01: Entry Domain and Domain02:Main Domain and can be seen in Figure while IEs include in each of them are presented in Figure and Figure .

P5-BP2-DA2 Definition of roles and access permissions

This activity involved identifying key stakeholders (both users and machines), as well as the resources that need to be exposed, and specifying the required permissions. These permissions were defined as a combination of actions that users can perform on the identified resources. The process considered both standard **aerOS** security requirements and the specific needs of the pilot scenario.

According to the standard **aerOS** design, HLOs and Data Fabric components across different domains should be provisioned with the necessary permissions to request remote deployments and retrieve domain capabilities and availability, respectively.

External actors—such as admins, users requesting service deployments, or those retrieving continuum data—were identified as required roles based on anticipated use cases. These use cases included IoT service management and orchestration, data acquisition to enable external applications to access IoT data, and continuum data acquisition for external applications that may require deployment requests (e.g., a User Plane Function (UPF)) to support 5G communication within the continuum or a cognitive scheduler to deploy a trust function) or build new functionalities on top of this data.

To fulfill these requirements, the following roles and their respective permissions were defined:

- **Continuum Administrator**: A role for users who have full access to the continuum. Users with this role can manage other users (create / delete) and their permissions, access all continuum services across domains (e.g., request deployments), and retrieve all registered data (both continuum and IoT data).
- **aerOS User**: A role that allows users to consume data from their domain but prevents them from making any changes within the domain.

- **Data Product Owner**: A role for users assigned to create data sources that will be integrated into the domain.
- **External User**: A role assigned to actors who can retrieve continuum information to assess the possibility of requesting service deployments or IoT data to build new applications.
- **Vertical Deployer**: A role for users who want to deploy new IoT applications or submit requests for deploying applications that extend or support their existing functionality.

These roles are found to support use cases which could be supported from aerOS within the OTE Smart building activities. The following descriptions present such cases and how the integration of these roles provide and controls access to the needed functionalities.

eploying IoT Applications

The **Vertical Deployer** role is essential for managing the deployment of IoT applications within **OTE's** system. In a typical use case, a **Vertical Deployer** may initiate the deployment of an IoT application to monitor HVAC systems in a building. This role involves making decisions to submit deployment requests to **aerOS**, ensuring that the appropriate resources and configurations are provisioned. By introducing this role, **OTE** validates that **aerOS MetaOS** can support the dynamic deployment of services across its continuum, empowering **OTE** to manage IoT applications that facilitate automation and data collection. The **Vertical Deployer**'s ability to request deployments ensures that **OTE** has control over the IoT ecosystem, enabling scalable and automated services.

Home Assistant Integration

In this scenario, Home Assistant, already deployed on-premises within **OTE**'s infrastructure, acts as an **application actor** within **aerOS MetaOS**. It interacts with the **aerOS** environment using the system's APIs to retrieve data, automate building functions (e.g., lighting control), and push updated data back into the system. The **aerOS User** role is essential here because it enables users to access the data exposed by **Home Assistant**, such as sensor readings, without granting them permissions to modify the system.

Data Product Creation

The **Data Product Owner** role becomes critical when data, such as that retrieved from **Home Assistant** or IoT sensors, needs to be transformed into actionable, consumable products. For example, a **Data Product Owner** might create a data product based on environmental sensor data (e.g., temperature, humidity) that can be accessed by other users or applications. This role ensures that raw IoT data is structured and made available for consumption by other users or services within the continuum. This functionality validates **OTE**'s ability to create data products and use them for decision-making processes. The **Data Product Owner** role ensures that **OTE** can not only manage raw data but also convert it into structured formats that can be easily accessed and used by **aerOS Users** or external systems, driving operational efficiency.

External User Access to Data for New Application Requests

An **External User**, such as a third-party developer, can retrieve data products created by the **Data Product Owner** to evaluate potential applications or services that can be built on top of this data. For example, the **External User** might analyze temperature data to determine whether a new energy optimization system is required. Based on the insights gained, the **External User** could request a new deployment from **OTE**'s system, such as a service that leverages the data for intelligent decision-making. This scenario validates that **OTE** supports external actors in not only consuming data but also requesting new applications and services based on that data. The **External User** role ensures that **OTE** can safely allow third-party actors to engage with the system, while maintaining control over data access and deployment requests.

Full System Control

The **Continuum Administrator** role oversees the entire **aerOS MetaOS** environment, with full authority over user management, access control, and system operations. This role is essential for ensuring the integrity and security of the system. The **Continuum Administrator** can grant, modify, or revoke user permissions across all domains and resources, ensuring that only authorized users can access sensitive data or perform critical tasks like deploying new services. The introduction of the **Continuum Administrator** role ensures that **OTE** has comprehensive control over its system, providing the necessary oversight to secure resources, monitor system activity, and enforce access policies across all levels of the continuum. This role validates that **OTE** can maintain system-wide security and governance while ensuring smooth operations and management.

The definition of these roles ensures that **OTE** can provide secure access to data, manage deployments, and integrate external actors into the ecosystem, all while maintaining a robust, flexible security framework. Each role is tailored to specific tasks, and together they create a dynamic system that supports **OTE**'s goals of efficient IoT management, secure data access, and scalable service deployment.

P5-BP2-DA3 Testbed deployment for 5g capabilities extension over aerOS

The development phase involved setting up the infrastructure necessary for the deployment of **5G network extensions** within the **aerOS MetaOS** framework. For this a new **aerOS domain** was created to support these network extensions, ensuring that the existing smart building functionalities remained unaffected, and the environment was isolated for security purposes. During this phase, **Open5GS** was deployed as the 5G core at **IP 10.220.2.73**, utilizing a dockerised version of **Open5GS** to interact with **Virtual Network Functions (VNFs)** across the continuum. This setup allowed for the seamless orchestration of 5G network functionalities while maintaining the integrity of other system domains. The following 2 screenshots present the testbed integrating one aerOS domain and an open5gs deployment. It is worth noticing that all needed VNFs are deployed except UPF which will be integrated in aerOS domain in the following steps.

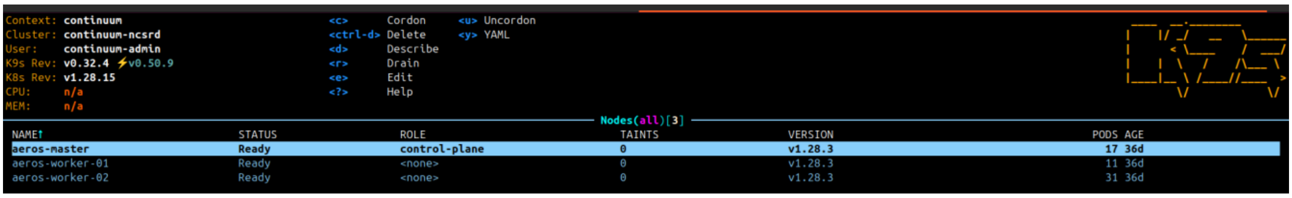


Figure 241. aerOS domain to support 5G extensions scenarios

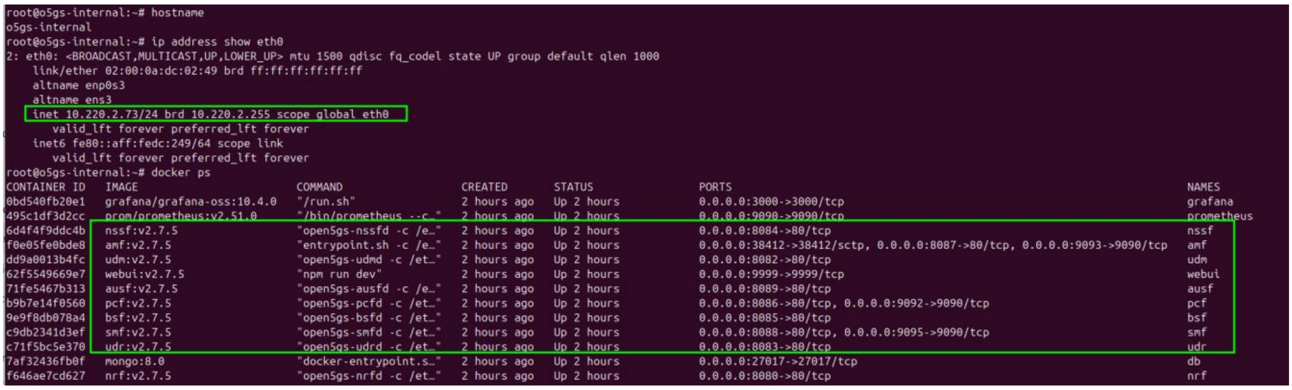


Figure 242. open5gs deployment to validate interaction with aerOS continuum

P5-BP2-DA4 OpenCAPIF deployment for secure NEF exposure from aerOS

In addition to the **Open5GS** deployment, an **OpenCAPIF** server was also deployed at IP 10.220.2.46 (external to aerOS domain). The **OpenCAPIF server** serves as a critical component in securely exposing 5G core services and providing controlled access to the **5G core VNFs.** CAPIF services are exposed over an API, and a Postman collection is built to interact with this API, this collection is demonstrated in next figure. CAPIF is a 3GPP-standardized framework designed to secure access to core network functions and expose them to external applications and services. While deployment of services over aerOS continuum is controlled by aerOS cybersecurity components, at application level, e.g., NEF endpoints, controlled access remains a concern of the application. By deploying OpenCAPIF server, we are able to securely expose **NEF** access over the **aerOS continuum**, providing a standardized and secure method for external applications to interact with the 5G core. The standard procedure for this secure exposure is described in 3GPP technical specification document: Common API Framework for 3GPP Northbound APIs (3GPP TS 23.222 version 18.7.0 Release 18)[22]

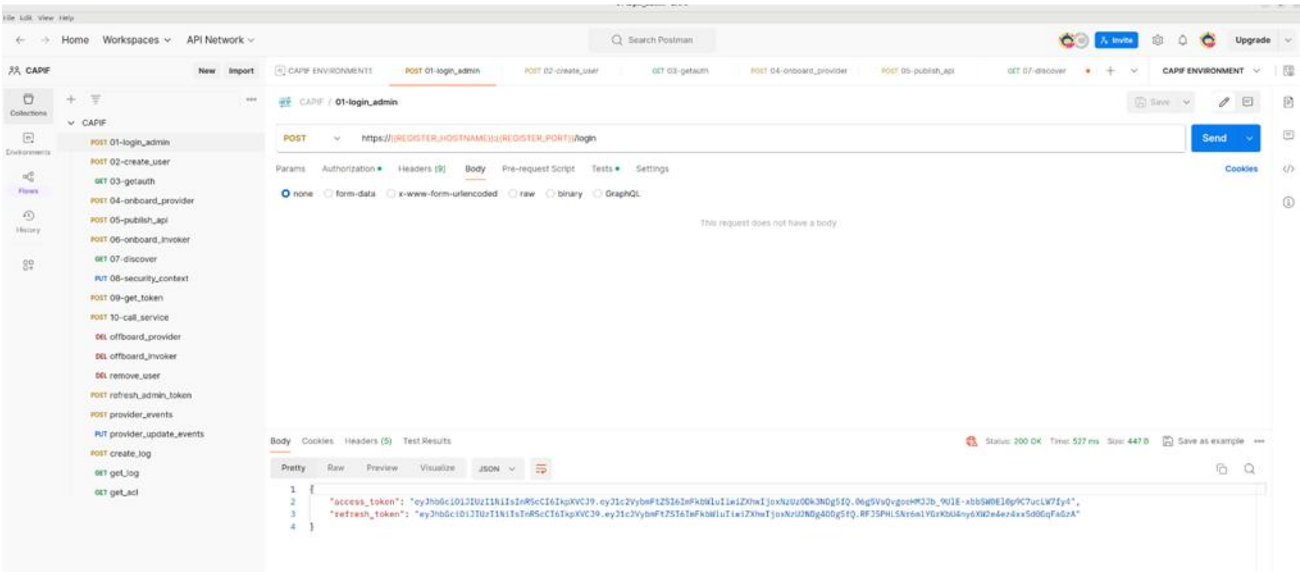


Figure 243. Postman collection for OpenCAPIF

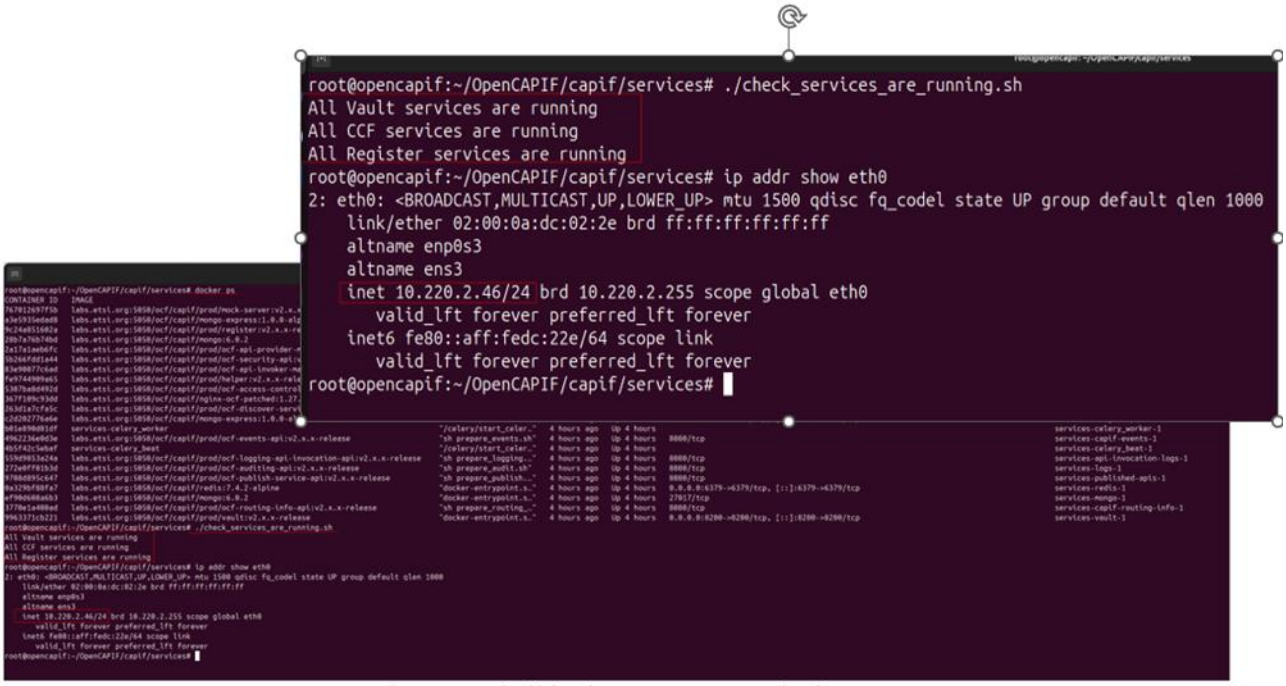


Figure 244. Shell in the OpenCAPIF deployment

P5-BP2-DA5 Adaptation of open5gs UPF function for aerOS onboarding

Open5gs is an open-source implementation of a 5G core, available in different deployment flavors, including monolithic and Docker-based deployments. Since aerOS is a MetaOS that orchestrates and manages containerized workloads, the Docker-based version of Open5gs is a natural fit for integration. The goal is to build a Docker image of Open5gs UPF that can be used for onboarding into aerOS, using HLO APIs. The whole process was carried out in a development server and produced a UPF image ready to be deployed on aerOS and wait a connection with an open5sg core deployed in the premises, under the condition being

[22] https://www.etsi.org/deliver/etsi_ts/123200_123299/123222/18.07.00_60/ts_123222v180700p.pdf

routable. The development process involves setting up the server environment (ensure docker and make tools are installed in the server) and building the Docker images using the following steps:

1 Clone the Open5gs Docker repository from GitHub: "git clone https://github.com/Borjis131/docker-open5gs.git"
2 Use "make" to build the base Open5gs image: "make base-open5gs"
3 Edit an adapted yaml file including information about intended UPF configuration
4 Build the new image that includes both the UPF container and the pre-configured YAML file: "docker build -t open5gs-upf-with-yaml"

The output is a UPF docker image including its needed configuration and ready to run over aerOS orchestrated continuum. This can be pointed by an open5gs and server as an entry point for the continuum.

P5-BP2-DA6 Adaptation of open5gs NEF function for aerOS onboarding

NEF is an exposure function which can securely provide access to selected resources of an 5g core. Although NEF functionalities are extensively documented in 3GPP documents (TS29.122, TS29.522) most cores have partial or no implementations for them. Open5gs is among these that do not provide such an implementation, and the NCSRD FRONT team has proceeded to implement basic functionalities that are of interest. The chosen functionalities are Event monitoring for UE location and session QoS. While these functionalities were not a dedicated effort within aerOS and only, the chance to validate an integration in the continuum was supported. The NEF capability integrated within aerOS was the UE location event monitoring. This functionality provides an interface with the information about the location of UE as recorded and hosted within the 5g core and supports updates with notifications to interested parties (network applications), which subsequently can make decisions based on this information.

The hosting repository for this NEF application hosted on GitHub[23]

To integrate the application within aerOS a TOSCA file needs to be carefully crafted pointing to the images repository and including the needed environment variables.

2.5.2.1.3. Integration activities

P5-BP2-IA1 Integration of aerOS cyber security services

As part of the process described in P5-BP1-IA15 "Integration of aerOS Basic IE Components," alongside all the aerOS basic components, the essential elements for enforcing cybersecurity across the continuum were deployed. The aerOS security architecture, as outlined in the WP3 deliverables, integrates several components:

LDAP: Serving as the users' registry engine, LDAP is responsible for managing and storing user information and authentication data. It acts as the **Identity Store**, ensuring user identity validation across the system.

Keycloak: This is the token provisioning engine, responsible for handling **authentication** and **authorization**. Keycloak provides centralized user identity management, and issues tokens for secure communication. It supports the use of **Access Control Lists (ACLs)** and **roles** to govern user permissions, enabling Single Sign-On (SSO) and federated identity management.

KrakenD: Serving as the **API Gateway**, KrakenD is the policy control and **Policy Enforcement Point (PEP)** in the system. It ensures that security policies are enforced at the gateway level, controlling access to microservices based on the tokens issued by Keycloak. KrakenD acts as the enforcement point for the defined policies, validating requests against authorization rules before forwarding them to backend services. It operates at the edge, enforcing the security policies across each domain in the continuum.

[23] https://github.com/FRONT-research-group/NEF

While **LDAP** and **Keycloak** typically have a **single instance** deployed across the continuum for identity and authentication management, **KrakenD** operates with a running instance in each aerOS domain to ensure that policy enforcement is localized and consistent for every part of the infrastructure.

The figures below show the deployment of these cyber security components and some access to their provided interfaces.

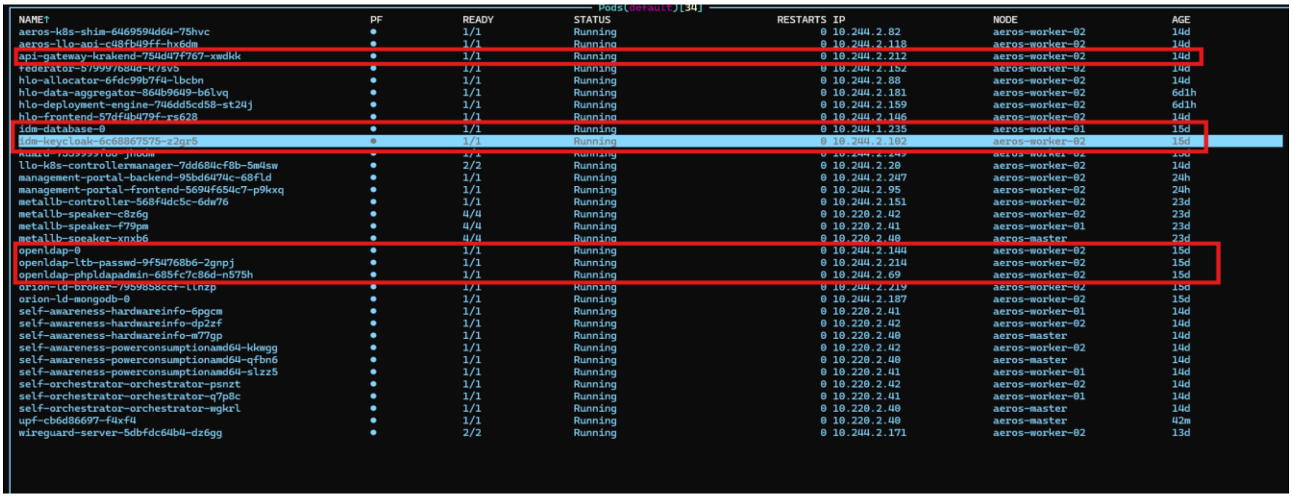


Figure 245. Deployment of aerOS cyber security components

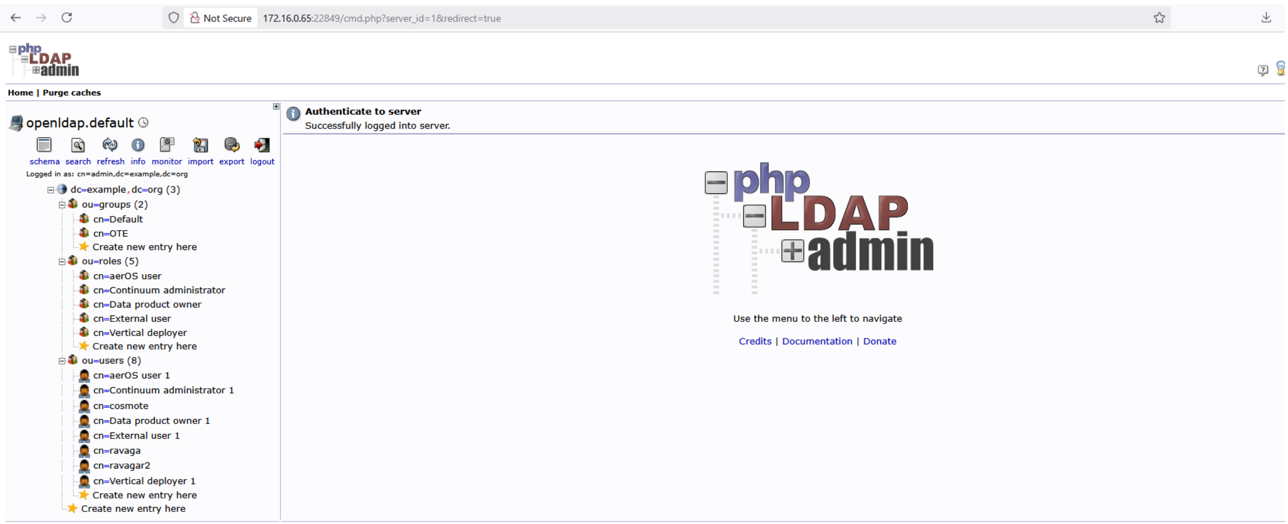


Figure 246. Access to the deployed LDAP to register actors

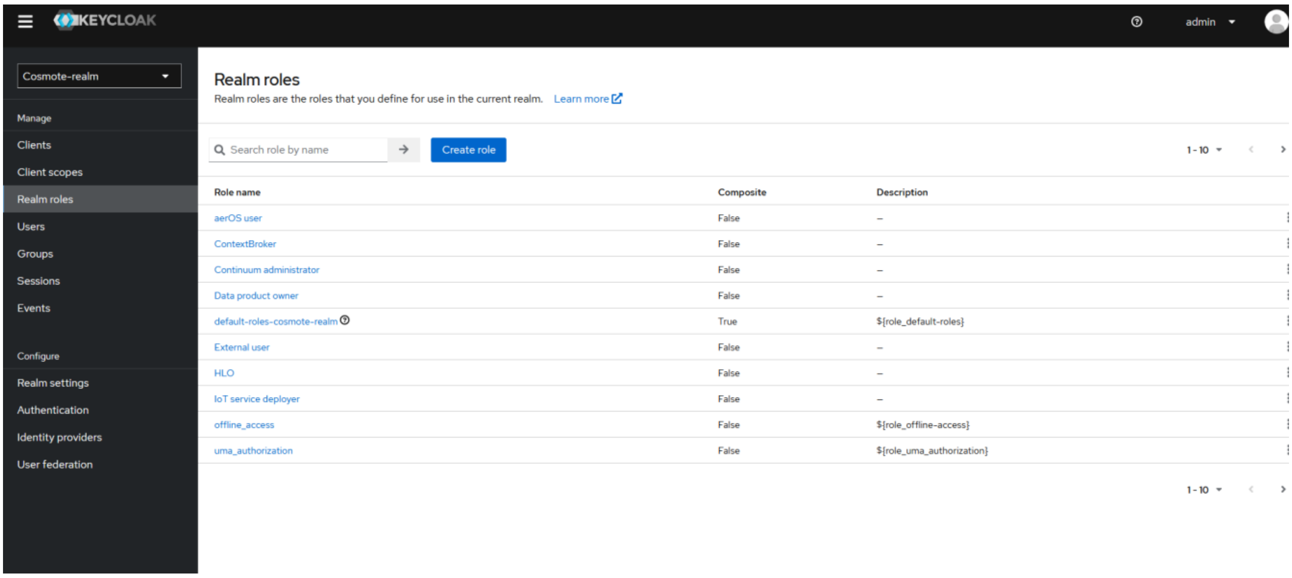


Figure 247. Access to the deployed Keycloak to make roles and define access permissions

P5-BP2-IA2 Integration of roles and permissions for aerOS APIs access

In this activity, we defined and integrated roles and permissions within **Keycloak** to manage controlled access to **aerOS APIs**. The roles and permissions were carefully configured in **Keycloak** to ensure that only authorized users and systems could interact with the system's resources. These configurations were designed to enforce security policies that restrict access based on the user's role, thus providing a robust access control mechanism for **aerOS**. The integrated rules in **Keycloak** safeguard access to the underlying resources, ensuring that each role has the appropriate level of permissions. Additionally, the **KrakenD API Gateway** was configured to enforce these access rules, ensuring secure and efficient interactions between clients and backend services. This integration between **Keycloak** and **KrakenD** guarantees that **aerOS** APIs are accessed securely, with all requests properly authenticated and authorized according to the defined role-based policies.

```
{
  "endpoint": "/dataProducts/{data_product_id}",
  "method": "GET",
  "output_encoding": "no-op",
  "extra_config": {
    "auth/validator": {
      "alg": "RS256",
      "roles_key": "realm_access.roles",
      "roles_key_is_nested": true,
      "roles": [
        "Continuum administrator",
        "Data product owner",
        "Data consumer"
      ],
      "jwk_url": "https://keycloak.front-research-group.eu/auth/realms/NCSRD/protocol/openid-connect/certs",
      "disable_jwk_security": true
    }
```

```
{
  "endpoint": "/orionld/ngsi-ld/v1/entities/{entityId}",
  "method": "GET",
  "output_encoding": "no-op",
  "extra_config": {
    "auth/validator": {
      "cache": true,
      "cache_duration": 3600,
      "alg": "RS256",
      "roles_key": "realm_access.roles",
      "roles_key_is_nested": true,
      "roles": [
        "Domain administrator",
        "Continuum administrator",
        "ContextBroker"
      ],
      "jwk_url": "https://keycloak.front-research-group.eu/auth/realms/NCSRD/protocol/openid-connect/certs",
      "disable_jwk_security": true
    }
```

```
{
  "endpoint": "/hlo_al/services/{service_id}/service_components/{service_component_id}/",
  "method": "GET",
  "output_encoding": "no-op",
  "extra_config": {
    "auth/validator": {
      "alg": "RS256",
      "roles_key": "realm_access.roles",
      "roles_key_is_nested": true,
      "roles": [
        "Continuum administrator",
        "IoT service deployer",
        "HLO"
      ],
      "jwk_url": "https://keycloak.front-research-group.eu/auth/realms/NCSRD/protocol/openid-connect/certs",
      "disable_jwk_security": true
    }
```

Figure 248. Configured policies in KrakenD enforcing Keycloak access permission rules

P5-BP2-IA3 Integration of UPF VNF in aerOS continuum

The User Plane Function (UPF) was deployed as an aerOS service within the dedicated aerOS domain. To ensure seamless integration, the aerOS orchestrator was used to preselect a specific IE (Integrated Element), which is essential for knowing in advance the IP address of the host IE. For this deployment, IP 10.220.2.40 was selected as the IE IP. This selection was critical for configuring the Open5GS core, to ensure proper communication with the UPF VNF. The deployment, with the appropriate policy in place, allowed the UPF to advertise itself at the IP of the chosen IE (10.220.2.40). Meanwhile, the Open5GS deployment, located on an external server at IP 10.220.2.73, was configured to communicate with the UPF and other VNFs deployed across the continuum. This integration ensured that the UPF could function securely within the aerOS domain, extending 5G network capabilities at the edge, while maintaining smooth and secure communication with the Open5GS core network.

In the next figure, a TOSCA deployment file (as expected from the aerOS HLO endpoint) is presented. This file conveys all the necessary configuration and deployment information for this integration. Although the deployment is performed via the portal, this TOSCA file (which in any case is what the portal produces towards the HLO API) explains all details at a glance.

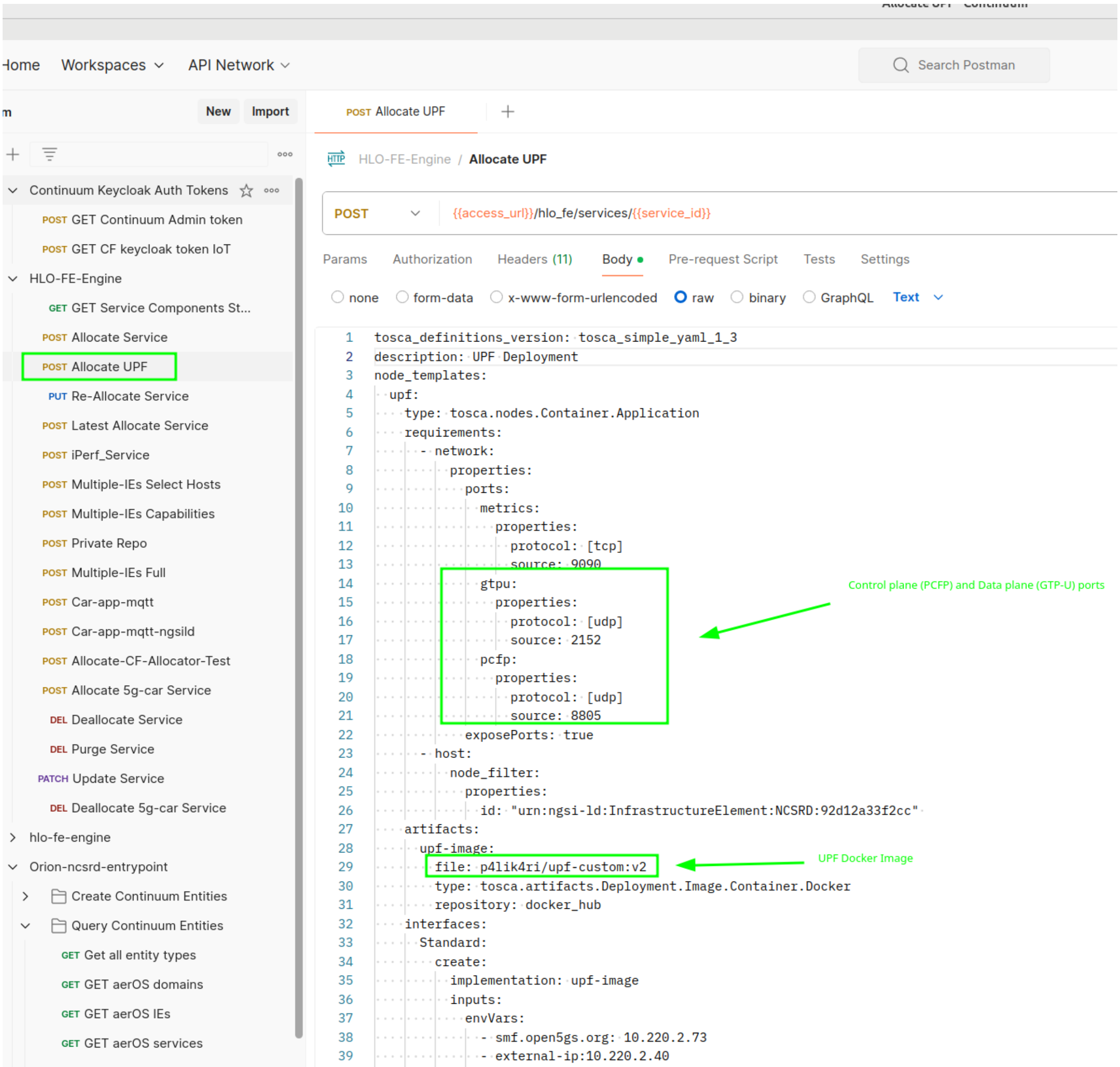


Figure 249. open5gs UPF deployment within aerOS continuum

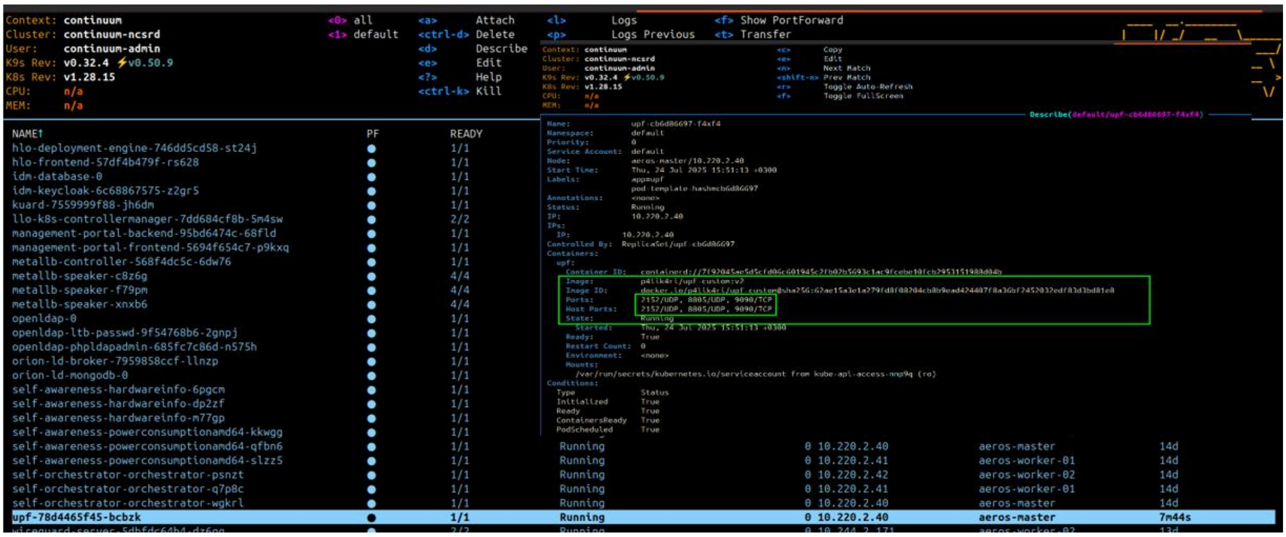


Figure 250. UPF deployed within aerOS continuum

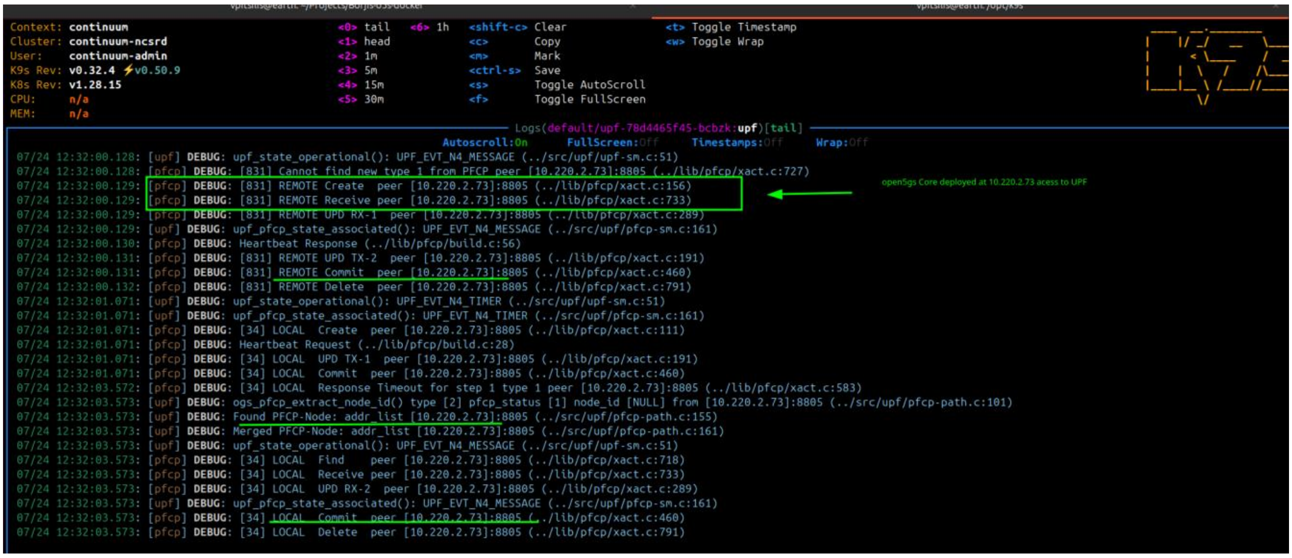


Figure 251. UPF aerOS deployment logs, showing the connection from 5g core

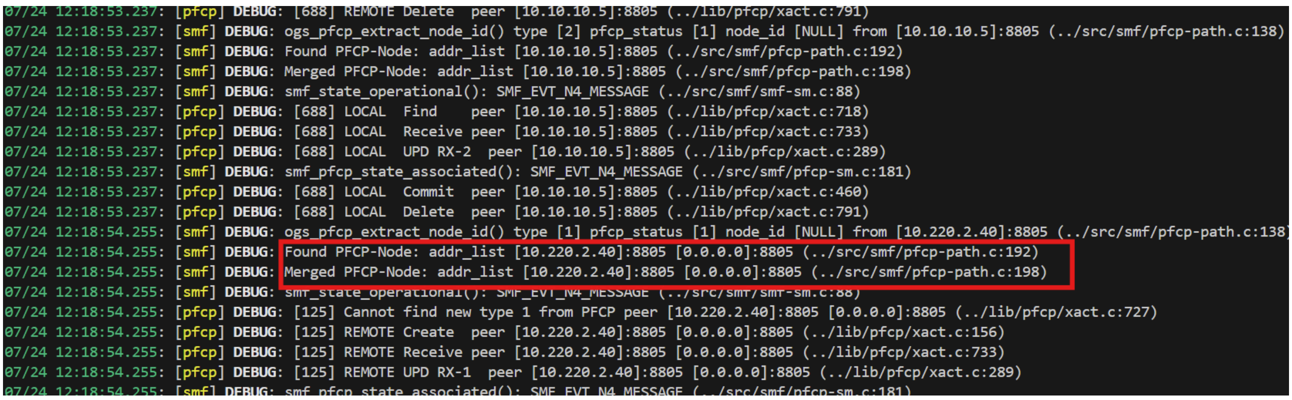
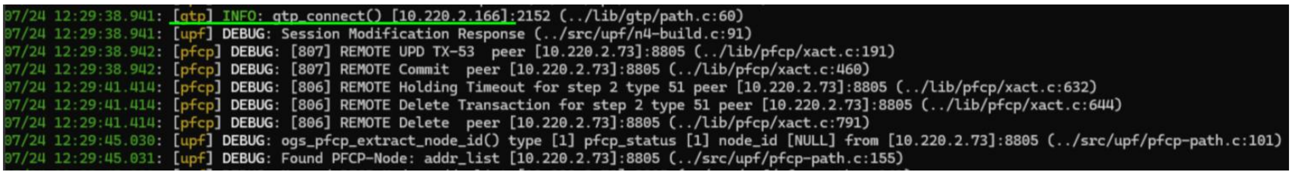


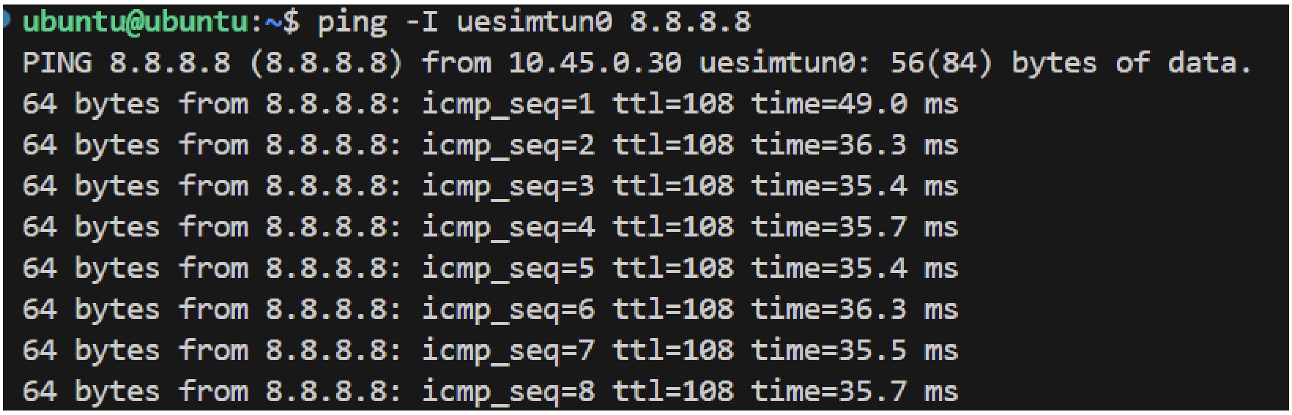
Figure 252. open5gs SMF logs, showing PFCP (control plane) access to UPF

Figure 2: UERANSIM establishing PDU session



```
07/24 12:29:38.941: [gtp] INFO: gtp_connect() [10.220.2.166]:2152 (../lib/gtp/path.c:60)
07/24 12:29:38.941: [upf] DEBUG: Session Modification Response (../src/upf/n4-build.c:91)
07/24 12:29:38.942: [pfcp] DEBUG: [807] REMOTE UPD TX-53  peer [10.220.2.73]:8805 (../lib/pfcp/xact.c:191)
07/24 12:29:38.942: [pfcp] DEBUG: [807] REMOTE Commit  peer [10.220.2.73]:8805 (../lib/pfcp/xact.c:460)
07/24 12:29:41.414: [pfcp] DEBUG: [806] REMOTE Holding Timeout for step 2 type 51 peer [10.220.2.73]:8805 (../lib/pfcp/xact.c:632)
07/24 12:29:41.414: [pfcp] DEBUG: [806] REMOTE Delete Transaction for step 2 type 51 peer [10.220.2.73]:8805 (../lib/pfcp/xact.c:644)
07/24 12:29:41.414: [pfcp] DEBUG: [806] REMOTE Delete  peer [10.220.2.73]:8805 (../lib/pfcp/xact.c:791)
07/24 12:29:45.030: [upf] DEBUG: ogs_pfcp_extract_node_id() type [1] pfcp_status [1] node_id [NULL] from [10.220.2.73]:8805 (../src/upf/pfcp-path.c:101)
07/24 12:29:45.031: [upf] DEBUG: Found PFCP-Node: addr_list [10.220.2.73]:8805 (../src/upf/pfcp-path.c:155)
```

Figure 253. UPF logs showing the GTP-U (data plane)



```
ubuntu@ubuntu:~$ ping -I uesimtun0 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 10.45.0.30 uesimtun0: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=108 time=49.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=108 time=36.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=108 time=35.4 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=108 time=35.7 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=108 time=35.4 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=108 time=36.3 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=108 time=35.5 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=108 time=35.7 ms
```

Figure 254. Sending from emulated UE (UERANSIM) to the internet through continuum based UPF

```
root@aeros-master:/# tcpdump -i ogstun host 10.45.0.30 and icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ogstun, link-type RAW (Raw IP), snapshot length 262144 bytes
12:36:48.135731 IP 10.45.0.30 > dns.google: ICMP echo request, id 154, seq 1, length 64
12:36:48.169985 IP dns.google > 10.45.0.30: ICMP echo reply, id 154, seq 1, length 64
12:36:49.136409 IP 10.45.0.30 > dns.google: ICMP echo request, id 154, seq 2, length 64
12:36:49.170528 IP dns.google > 10.45.0.30: ICMP echo reply, id 154, seq 2, length 64
12:36:50.137886 IP 10.45.0.30 > dns.google: ICMP echo request, id 154, seq 3, length 64
12:36:50.172094 IP dns.google > 10.45.0.30: ICMP echo reply, id 154, seq 3, length 64
12:37:03.657860 IP 10.45.0.30 > dns.google: ICMP echo request, id 155, seq 1, length 64
12:37:03.691960 IP dns.google > 10.45.0.30: ICMP echo reply, id 155, seq 1, length 64
12:37:04.662315 IP 10.45.0.30 > dns.google: ICMP echo request, id 155, seq 2, length 64
12:37:04.696386 IP dns.google > 10.45.0.30: ICMP echo reply, id 155, seq 2, length 64
12:37:05.657398 IP 10.45.0.30 > dns.google: ICMP echo request, id 155, seq 3, length 64
12:37:05.691480 IP dns.google > 10.45.0.30: ICMP echo reply, id 155, seq 3, length 64
12:37:06.658280 IP 10.45.0.30 > dns.google: ICMP echo request, id 155, seq 4, length 64
12:37:06.692363 IP dns.google > 10.45.0.30: ICMP echo reply, id 155, seq 4, length 64
12:37:07.658714 IP 10.45.0.30 > dns.google: ICMP echo request, id 155, seq 5, length 64
12:37:07.692821 IP dns.google > 10.45.0.30: ICMP echo reply, id 155, seq 5, length 64
12:37:08.660406 IP 10.45.0.30 > dns.google: ICMP echo request, id 155, seq 6, length 64
12:37:08.694475 IP dns.google > 10.45.0.30: ICMP echo reply, id 155, seq 6, length 64
12:37:09.665331 IP 10.45.0.30 > dns.google: ICMP echo request, id 155, seq 7, length 64
12:37:09.699419 IP dns.google > 10.45.0.30: ICMP echo reply, id 155, seq 7, length 64
12:37:10.665338 IP 10.45.0.30 > dns.google: ICMP echo request, id 155, seq 8, length 64
12:37:10.699472 IP dns.google > 10.45.0.30: ICMP echo reply, id 155, seq 8, length 64
```

Figure 255. Tcpdump on the UPF side

P5-BP2-IA4 Integration of NEF VNF in aerOS continuum

In this activity, we successfully integrated the **Network Exposure Function (NEF)** VNF within the **aerOS continuum**. The integration process was carried out using the **aerOS portal**, where we provided all the necessary configuration variables to enable the **NEF** function to connect to the **Open5GS server** and retrieve vital information, such as the reporting of **UE location**. The **TOSCA deployment file** is presented in , which outlines all the deployment details. Although the requested file was generated directly from the **aerOS portal**, we present it here to illustrate the specific details of the deployment request.

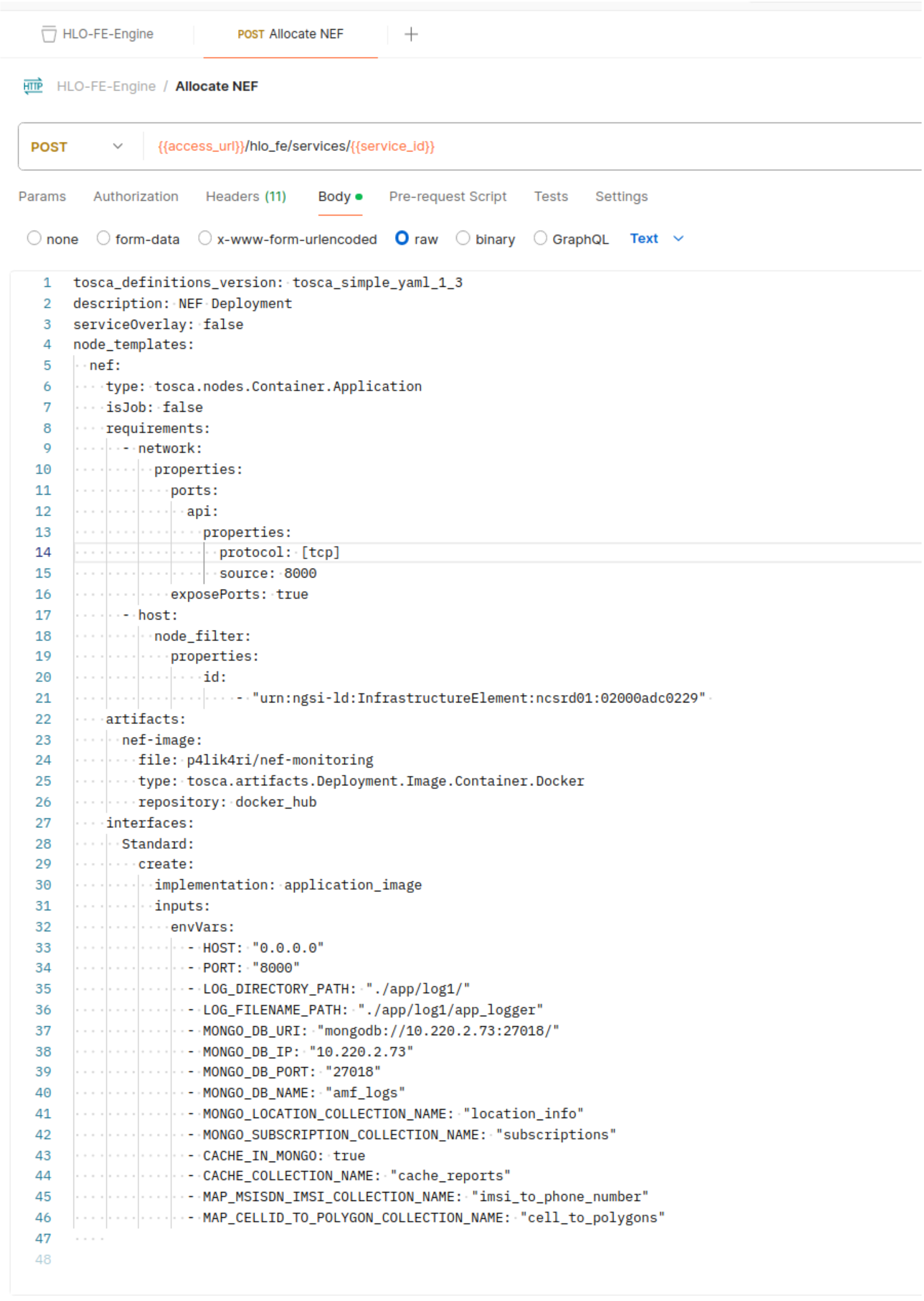


Figure 256. TOSCA file for the NEF deployment

Additionally, the next set of screenshots shows the **HLO (High-Level Orchestration)** process involved in deploying the **NEF** within the **aerOS** environment. These figures offer an insight into how the **NEF** was successfully integrated and orchestrated within the **aerOS** framework.

Figure 257. aerOS HLO orchestrating NEF deployment

In the following figure, we display the results on the **aerOS dashboard**, where we can see the details of the deployed **NEF** service, including deployment details and its current status.

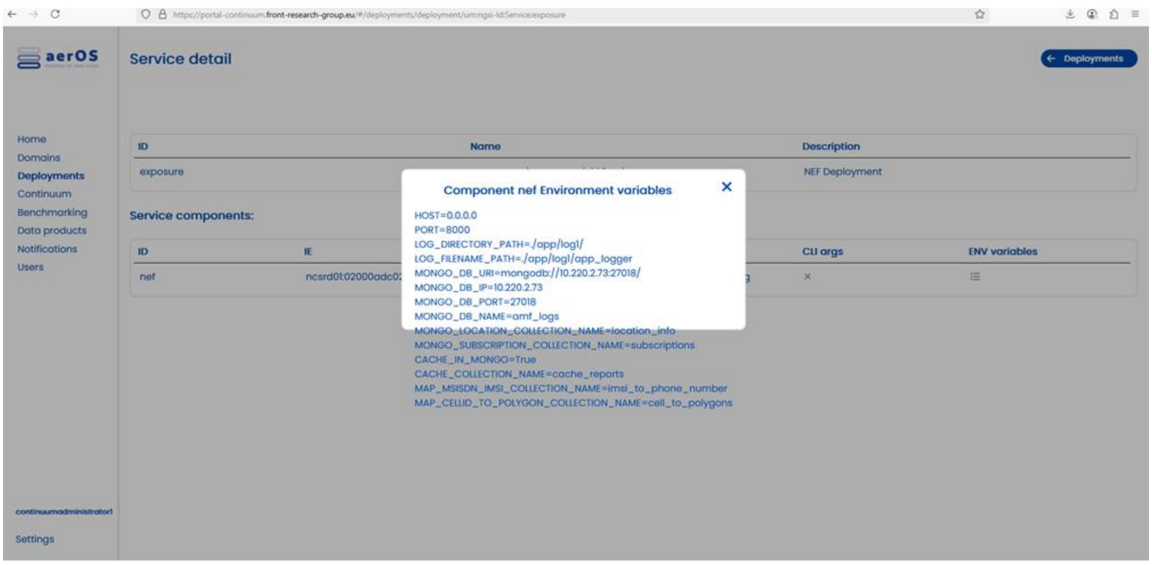


Figure 258. NEF service in the aerOS portal

Once deployed, the **NEF API** is accessible as an **aerOS managed service** and can be accessed via the exposed **node port**, as shown in **the following figure**. The **Swagger** interface for the API is also displayed below in the figure, providing a clear view of the available operations for interacting with the **NEF** service.

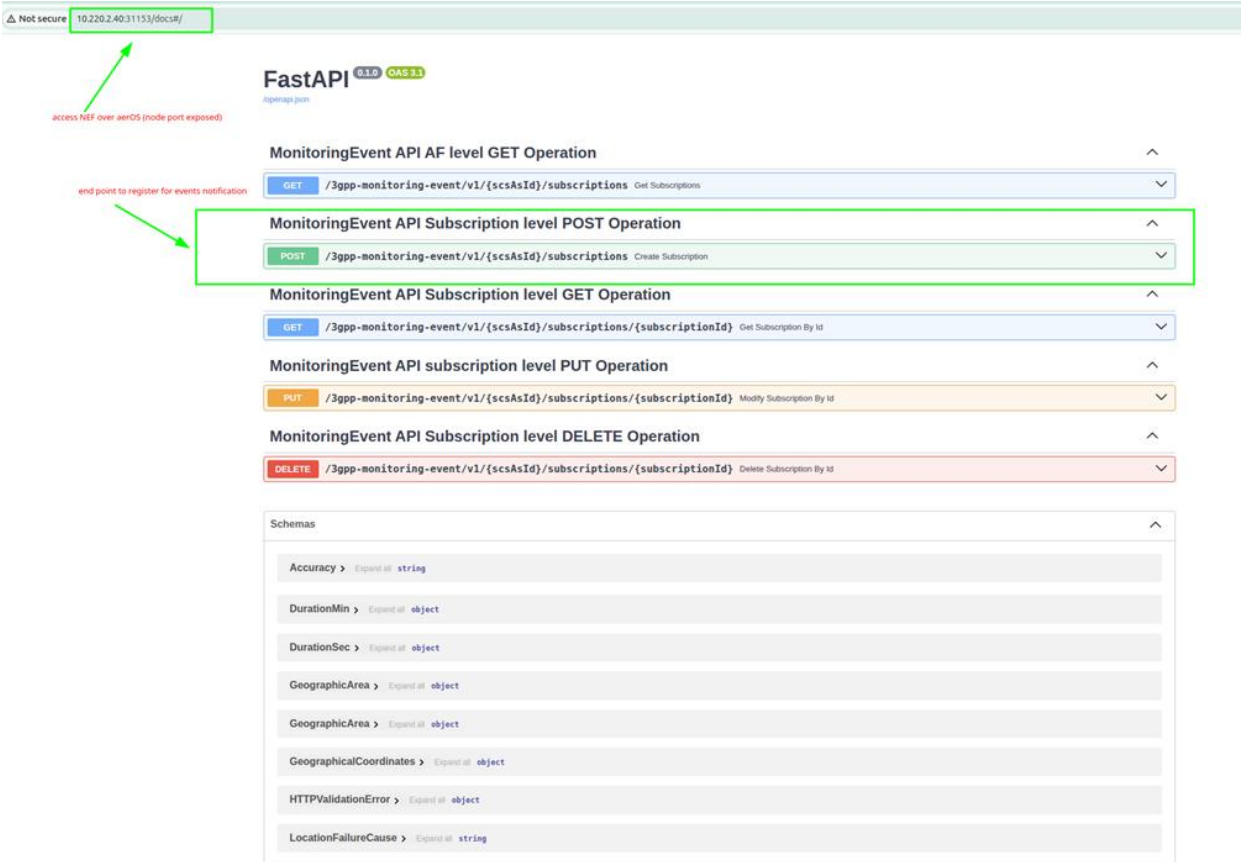


Figure 259. NEF API access over the continuum

Finally, in the subsequent screenshots, we demonstrate how the **NEF API** was used to register for **event reporting**. We successfully received three notifications, confirming the **NEF**'s functionality in reporting **UE**-related events. The data for the **UEs** were generated using **UERANSIM**, while the **UPF** continued to operate within the **aerOS continuum**, ensuring the seamless operation and reporting of 5G network functions.

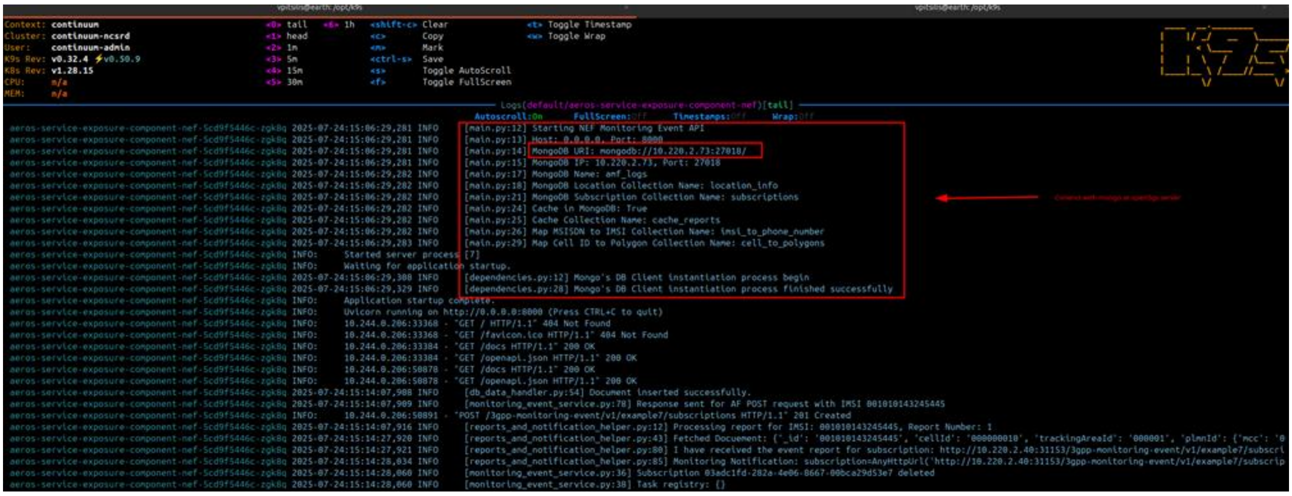


Figure 260. NEF aerOS service successfully connecting with service at open5gs server

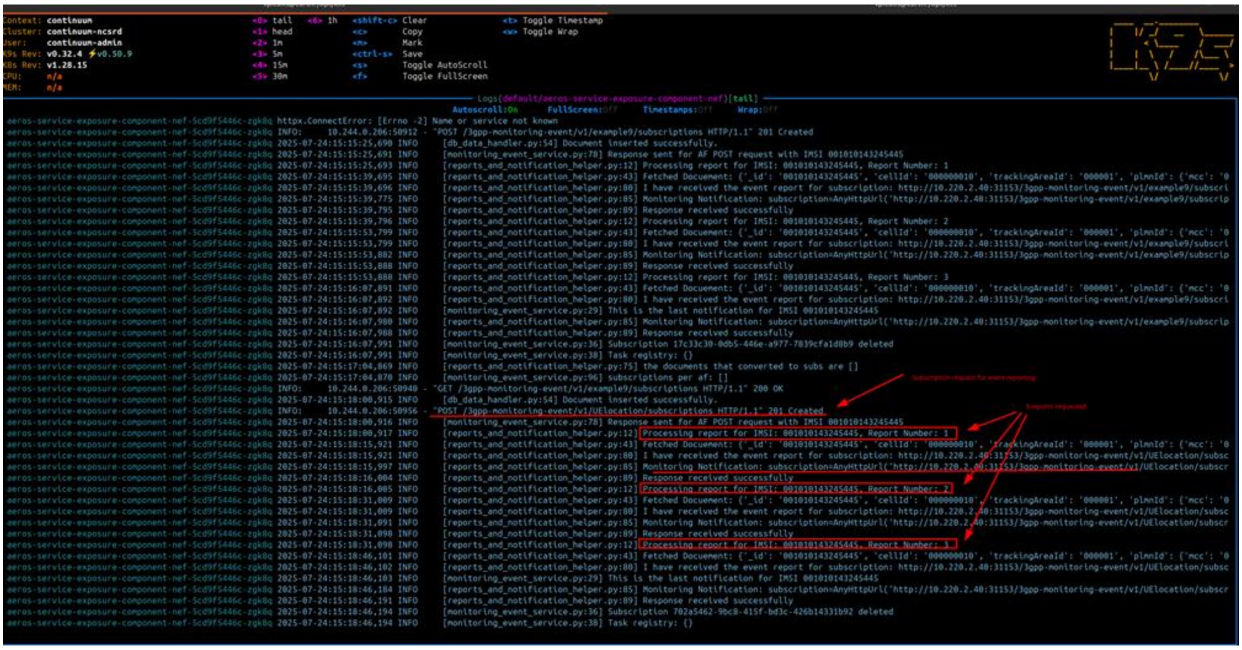


Figure 261. NEF monitoring event subscription invoked from exposed API and sends notifications

In conclusion, the topology presented in Figure is validated, as it demonstrates the successful integration of an external **Open5GS server** and the extension of the **5G core** by deploying both the **UPF** and **NEF** over the **aerOS continuum**. Despite the deployment being distributed across different network domains, we ensured that the 5G core was extended securely and that the system continued to interact properly with the **UPF** and **NEF** components. This demonstrates the promising potential of **aerOS MetaOS** in enabling flexible, scalable, and secure **5G network extensions** at the edge, offering further opportunities for future enhancements and integration with other 5G functionalities. The successful deployment and interaction of these components mark a key step in validating **aerOS** as a robust platform for extending **5G networks** in diverse environments.

P5-BP2-IA5 Register aerOS services to OpenCAPIF

In our deployment of the NEF over aerOS, we took an essential step towards enhancing service discoverability and secure access by registering the exposed APIs with OpenCAPIF. This registration process facilitates the ability of external invokers to discover and securely interact with the deployed services over aerOS, ensuring a seamless and secure API access layer.

To achieve this integration, we followed the official guidelines outlined in the ETSI documentation[24]. This comprehensive documentation provided the necessary steps to properly register a provider and enable the publication of the NEF notifications endpoint.

By registering the provider and exposing the NEF notifications endpoint, we enable external clients to discover, interact with, and securely invoke the available services hosted over aerOS, thus optimizing service exposure, and ensuring compliance with industry standards.

The procedure for this integration is depicted in the following figures. For the interaction with the OpenCAPIF API, the Postman collection referenced above was used.

As a first step we created an OpenCAPIF user.

[24] https://ocf.etsi.org/documentation/latest/

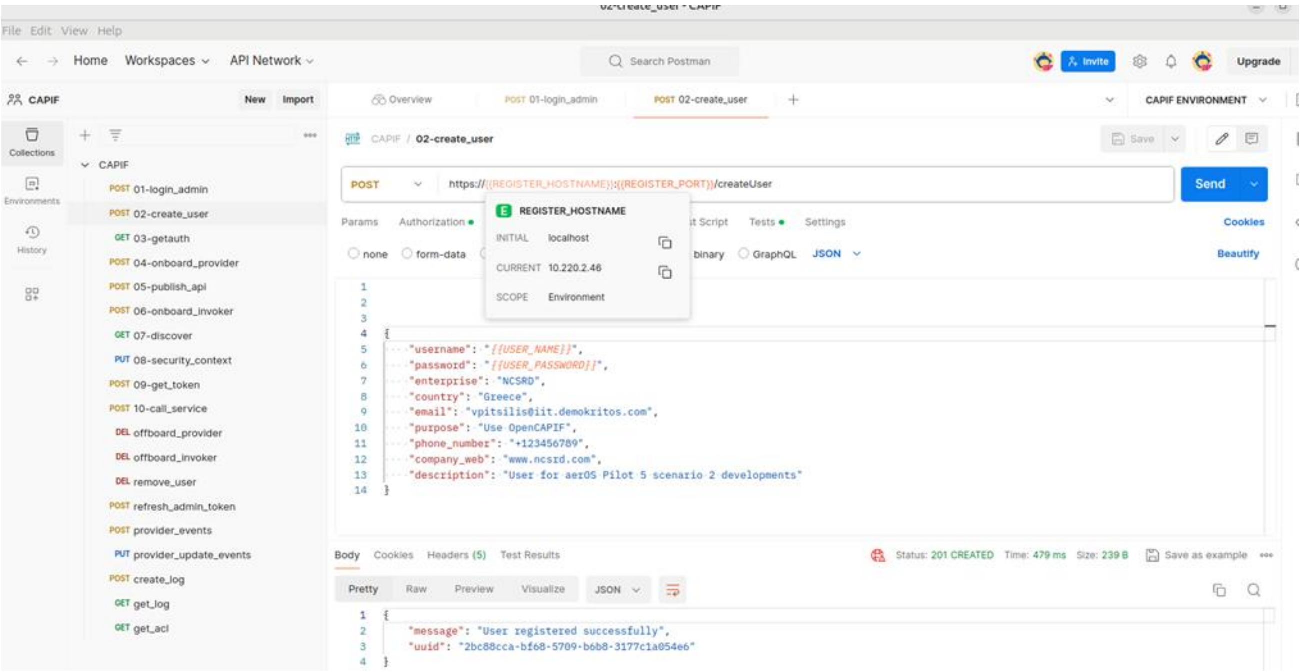


Figure 262. Registering OpenCAPIF user

aerOS OpenCAPIF user registered as an API provided. Upon registration PEM certificates and private and public kyes were created from OpenCAPIF server to be used for the next steps when publishing API endpoints. The respective service and keys creation is shown in the top left corner of the following figure.

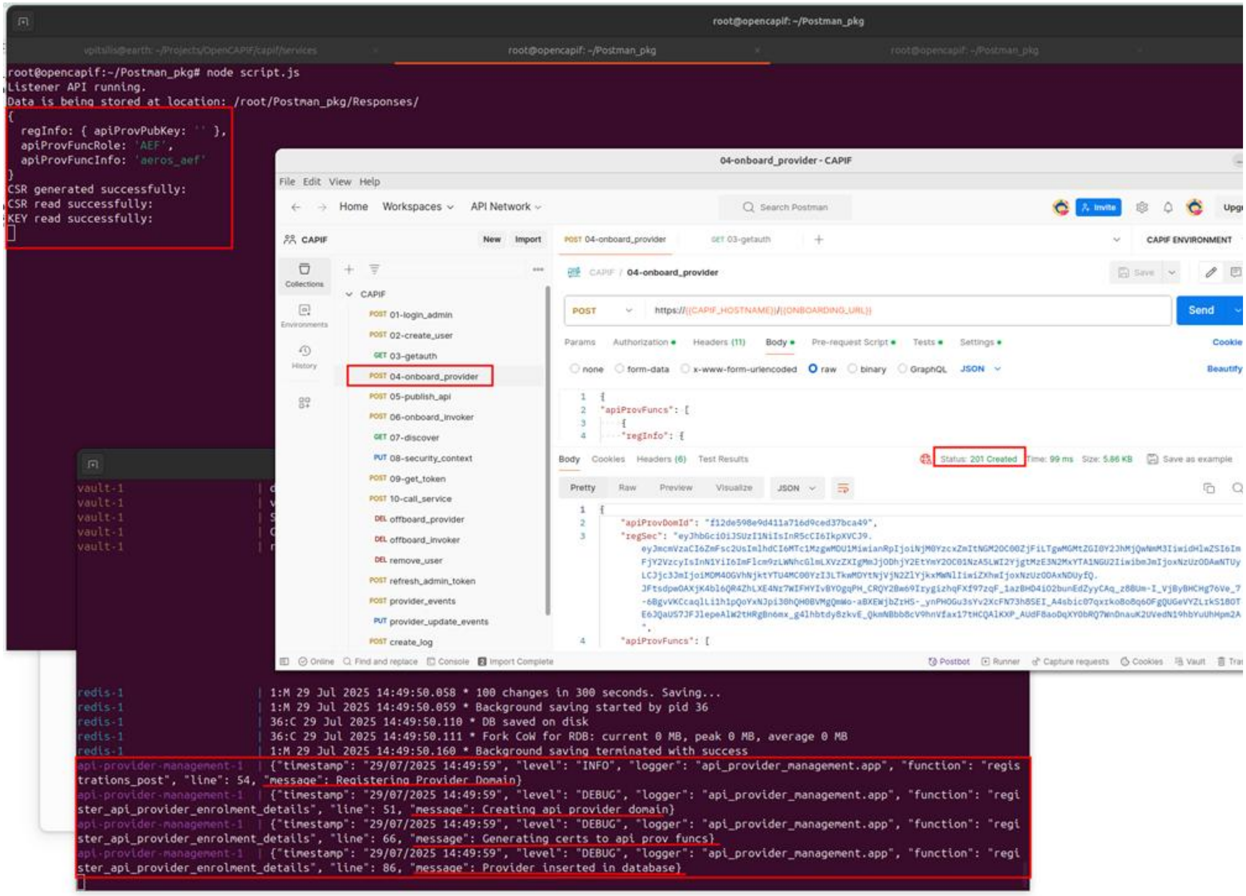


Figure 263. Registering API provider

Registered provider is publishing endpoint details, from now on NEF resource can be discovered from external invokers and accessed based on security policy rules.

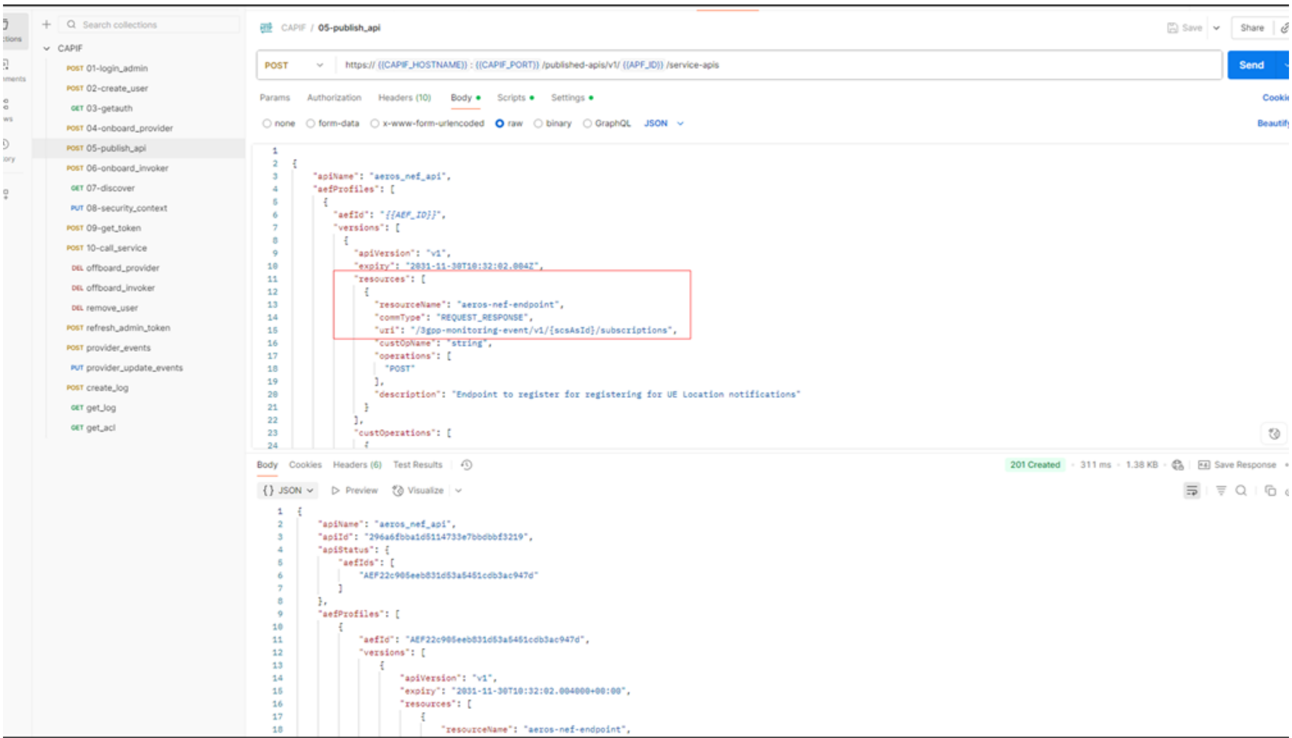


Figure 264. Registering NEF Endpoint deployed over aerOS

Finally, the following figure showcases the accessing of OpenCAPIF as a potential invoker and discovering exposed NEF API endpoint.

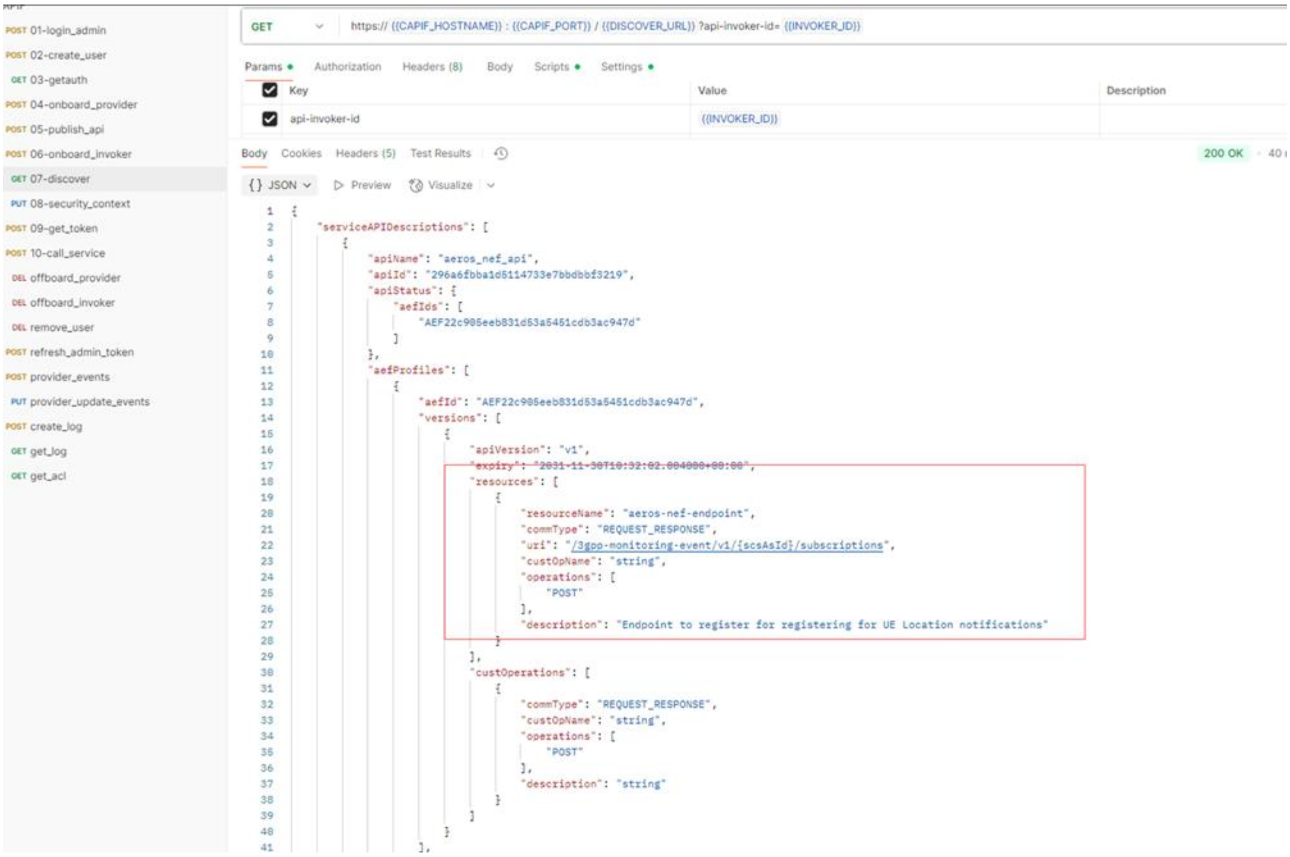


Figure 265. Discover NEF Endpoint deployed over aerOS

Subsequent NEF API invocation is intermediate, regarding trust and permissions control, from OpenCAPIF. Invoker acquires certificates and can access the exposed, over aerOS, NEF API with permissions already defined.

2.5.3. Pilot 5 Time-plan

The following Gantt diagram list all the activities from M19 (February 2024) until M35 (July 2025), representing evolution from status declared in D5.3.

Table 28. Pilot 5 updated Gantt timeline.

Pilot 5		2024											2025								
Code	**Name**	**M19**	**M20**	**M21**	**M22**	**M23**	**M24**	**M25**	**M26**	**M27**	**M28**	**M29**	**M30**	**M31**	**M32**	**M33**	**M34**	**M35**	**M36**	**M37**	**M38**
Business Process 1 (Scenario 1) - Smart Buildings - Intelligent Occupational Safety and Health																					
Setup & Procurement Activities																					
P5-BP1-SA1	Site survey for the Selection of Pilot5 Building/Rooms	completed by M18																			
P5-BP1-SA2	Procurement of Servers & Equipment	completed by M18																			
P5-BP1-SA3	Identification of Appropriate Smart Building Sensors	completed by M18																			
Development Activities																					
P5-BP1-DA4	Deployment & Maintenance of Sensors	completed by M18																			
P5-BP1-DA5	Deployment of the IoT backend/Home Assistant	completed by M18																			
P5-BP1-DA6	Installation of the aerOS Domains	completed by M18																			
P5-BP1-DA7	Transformation of the IoT backend as AerOS IE	completed by M18																			
P5-BP1-DA8	HVAC/Plugs Actuator Component																				
P5-BP1-DA9	Forecast Engine - Health Index AI Component																				
P5-BP1-DA10	Forecast Engine – Environmental AI Component																				
P5-BP1-DA11	Forecast Engine - Energy Efficiency AI Component																				
P5-BP1-DA12	Health and Energy Optimization																				
P5-BP1-DA13	Recommender																				
P5-BP1-DA14	End-user GUI Application																				
Integration Activities																					
P5-BP1-IA15	Integration of AerOS Basic (MVP) IE Components																				
P5-BP1-IA16	Integration of aerOS non Basic IE Components																				
P5-BP1-IA17	Integration of Data fabric with the IoT backend																				
P5-BP1-IA18	Integration of Data Fabric with Forecast Health Index System																				

P5-BP1-IA19	Integration of Data Fabric with Forecast Environmental AI System	
P5-BP1-IA20	Integration of Data Fabric with Forecast Energy Efficiency System	
P5-BP1-IA21	Integration of Data Fabric with Optimization System	
P5-BP1-IA22	Integration of Data Fabric with Recommender System	
P5-BP1-IA23	Integration of Data Fabric with End-user GUI System	
P5-BP1-IA24	E2E Integration of all Application Components (IoT, Forecast Engine, Recommender, GUI)	
Validation Activities		
P5-BP1-VA25	End-to-End Demonstrator (Seating Recommendation)	
P5-BP1-VA26	Pilot Services Created, Managed and Operated by AerOS Orchestrator	
P5-BP1-VA27	Energy Use Reduction	
P5-BP1-VA28	Edge Processing Performance Gains	
P5-BP1-VA29	Service Availability within the AerOS IE	
P5-BP1-VA30	Service Creation / Scalability	
P5-BP1-VA31	Improvement of Air Quality	
Business Process 2 (Scenario 2) - Cybersecurity and data privacy in building automation		
Setup & Procurement Activities		
P5-BP2-SA1	Procurement of Servers & Equipment	Completed by M18 (See use case 1)
P5-BP2-SA2	Identification of Targeted 5G Core	Completed by M18
Development Activities		
P5-BP2-DA1	Installation of aerOS Domains	Completed by M18
P5-BP2-DA2	Definition of roles and access permissions	
P5-BP2-DA3	Testbed deployment for 5g capabilities extension over aerOS	
P5-BP2-DA4	OpenCAPIF deployment for secure NEF exposure from aerOS	
P5-BP2-DA5	Adaptation of open5gs UPF for aerOS onboarding	
P5-BP2-DA6	Adaptation of open5gs NEF for aerOS onboarding	
Integration Activities		
P5-BP2-IA1	Integration of aerOS cyber security	Completed

	services	by M18																			
P5-BP2-IA2	Integration of roles and permissions for aerOS APIs access																				
P5-BP2-IA3	Integration of UPF VNF in aerOS continuum																				
P5-BP2-IA4	Integration of NEF VNF in aerOS continuum																				
P5-BP2-IA5	Register aerOS services to Open-CAPIF																				
Validation Activities																					
P5-BP2-VA1	5G E2E deployment validation with VNFs over aerOS (UERANSIM)																				
P5-BP2-VA2	Access control based on established RBAC rules																				

3. aerOS Components Integration Assessment

The following table provides a summary of the aerOS components deployed in each pilot. Detailed information on their integration across the different domains, including versioning and other technical specifications, is presented in the **Integration Activities** section of each pilot.

It can be observed that **all components of aerOS Meta-OS (both basic and non-basic) have been validated in, at least, one pilot (i.e., scenario of a pilot)**. Therefore, it can be determined that all the architecture, and the developments performed in WP3 and WP4 have proven useful for completing essential or auxiliary goals in the pilots of the project.

Table 29. aerOS components integration assessment

aerOS Component	P1.1	P1.2	P1.3	P1.4	P2	P3	P4	P5
WG for SC internal networking	X		X		X	X		X
Nginx Ingress Controller	X				X		X	X
Metal LB load balancer	X				X			X
Cert-Manager and TLS certificates	X				X		X	X
OpenAPI	X		X			X		X
AsyncAPI			X					
Low-code			X					
HLO Frontend Engine	X	X	X	X	X	X	X	X
HLO Data Aggregation and Alert System	X	X	X	X	X	X	X	X
HLO Allocation Engine	X	X	X	X	X	X	X	X
HLO Deployment Engine	X	X	X	X	X	X	X	X
LLO-K8s	X	X	X	X	X	X	X	X
LLO-Docker			X	X				
Keycloak	X	X	X	X	X	X	X	X
Kraken-D	X	X	X	X	X	X	X	X
Open LDAP	X	X	X	X	X	X	X	X
Self-awareness	X	X	X	X	X	X	X	X
Self-orchestration	X	X	X	X	X	X	X	X
Self-security		X			X	X	X	X
Self-healing		X				X		X
Self-realtimeness						X		
Self-adaptation						X		
Self-configuration					X			
Self-scaling	X							
Self-API	X					X		
Semantic Annotator								X
Semantic Translator								X
Continuum Ontology							X	

Orion-LD	X	X	X	X	X	X	X	X
Data Catalog						X		X
LDAP collector						X		X
Morph-KGC								X
RDF-NGSI-LD								X
Data Product Manager								X
AI distributed inference (AI Local Execution)						X		
AI distributed training (AI Local Execution + AI Task Controller)							X	X
Frugal – Model (quantiz., pruning)							X	
XAI - Shapley								X
EAT	X							X
Templates	X							X
Stratified Sampling	X	X						X
Anomaly Detection	X	X						
Data Drift	X							
Visualisation	X							
IOTA	X	X	X	X	X	X	X	X
Trust score						X	X	X
Management Portal	X	X	X	X	X	X	X	X
MP Frontend	X	X	X	X	X	X	X	X
MP Backend	X	X	X	X	X	X	X	X
Entrypoint Balancer			X			X		
Benchmarking Tool			X			X	X	X
Federator	X	X	X	X	X	X	X	X

4. Standards Used

This section provides an overview of the most relevant standards adopted across the aerOS pilot implementations. It focuses on standards that play a specific and meaningful role in the deployment or integration of aerOS technologies within each use case domain. Only domain-specific or implementation-relevant standards have been included; widely used foundational technologies such as **HTTP**, **JSON**, **REST** or similar others—while essential—are considered too generic to provide added analytical value and are therefore not listed.

Table 30. Standard NGSI-LD

Applied standard	ID and Title	**NGSI-LD** ID: ETSI ISG CIM NGSI-LD Title: Context Information Management (NGSI-LD) Specification
	Date of publication	2017
	Publisher	ETSI (European Telecommunications Standards Institute)
	License	**Public** – Open Standard
Pilot mapping	aerOS pilot	**Global aerOS Meta-OS, all pilots.**
	Technology name	Data processing (Orion-LD Context Broker, FIWARE)
	Usage specifics	NGSI-LD structures context data from the Stations, transformed in Node-RED, and sent to the Orion-LD Context Broker via PATCH requests for interoperable data management.
Gaps	Identified standardization gaps	No gaps identified
Application areas	Integration	X
	Communication	X
	IoT	X

Table 31. Standard OPC-UA

Applied standard	ID and Title	**OPC-UA** – Open Platform Communications Unified Architecture
	Date of publication	2008 (initial release) **Continuously updated** (e.g., Version 1.04 in 2020)
	Publisher	OPC UA Foundation
	License	**Public** - Open Standard
Pilot mapping	aerOS pilot	**Pilot 1.2, Pilot 1.3**
	Technology name	Use in multiple IoT and Edge equipment, to enabler secure and Intelligent measurement and calibration processes.
	Usage specifics	OPC UA is used to enable standardized, secure, and platform-independent communication between metrology devices, edge nodes, and calibration systems. It ensures semantic interoperability and structured data exchange within the calibration and measurement workflow.
Gaps	Identified standardization gaps	Some OPC UA models in metrology are not yet fully standardized or may require extensions. Interoperability challenges arise when integrating legacy systems lacking OPC UA support.

Application areas	
Integration	X
Communication	X
Information	X

Quality	X
Safety & Security	X
IoT	X

Table 32. Standard TSN

Applied standard	ID and Title	IEEE 802.1AS-2020 - **Time-Sensitive Networking**
	Date of publication	2020
	Publisher	IEEE
	License	Public
Pilot mapping	aerOS pilot	**Pilot 1.3**
	Technology name	Multiport Time Sensitive Networking (MTSN) Kit from SoCe
	Usage specifics	Implements deterministic Ethernet networking for real-time industrial communication
Gaps	Identified standardization gaps	
Application areas		
Integration		X
Communication		X
IoT		X

Table 33. Standard PROFINET

Applied standard	ID and Title	**PROFINET** IEC 61784-2 - Industrial communication networks - Profiles
	Date of publication	2003
	Publisher	IEC & PI (PROFIBUS & PROFINET International)
	License	Public
t n a	aerOS pilot	**Pilot 1.3**

	Technology name	AGVs, Robot Arm, Industrial Edge Components, PLCs
	Usage specifics	Main industrial ethernet communication protocol used for real-time data exchange between industrial automation components
Gaps	Identified standardization gaps	None significant identified

Application areas	
Integration	X
Communication	X
IoT	X

Table 34. Standard PROFIsafe

Applied standard	ID and Title	**PROFIsafe** IEC 61784-3-3 - Functional safety communication profiles
	Date of publication	2005
	Publisher	IEC & PI (PROFIBUS & PROFINET International)
	License	Public
Pilot mapping	aerOS pilot	**Pilot 1.3**
	Technology name	Safety PLCs, Safety Laser Scanners, Safety-related components
	Usage specifics	Safety protocol running on top of PROFINET for secure communication between safety components
Gaps	Identified standardization gaps	None significant identified

Application areas	
Integration	X
Communication	X
Safety & Security	X

Table 35. Standard DDS

Applied standard	ID and Title	**DDS (Data Distribution Service)** Object Management Group (OMG) DDS Specification v1.4
	Date of publica-tion	2015
	Publisher	Object Management Group (OMG)
	License	Public
Pilot mapping	aerOS pilot	**Pilot 1.3**
	Technology name	AGVs, Robot Arm and, Industrial Edge Components (via ROS 2 implementa-tion)
	Usage specifics	Provides real-time, publish-subscribe data distribution standard used as the middleware for ROS 2, enabling reliable communication between robotic components
Gaps	Identified standardization gaps	None significant identified in the core standard
Application areas		
Integration		X
Communication		X
IoT		X

Table 36. Standard TIC4.0 CHE Data Model

Applied standard	ID and Title	**TIC 4.0 CHE Data Model 2025.015** Addition information
	Date of publication	13/06/2025
	Publisher	TIC 4.0 Committee
	License	Public
Pilot mapping	aerOS pilot	**Pilot 4**
	Technology name	CHE time-series data
	Usage specifics	The primary purpose of TIC4.0 is to enable interoperability among

		equipment, systems, and stakeholders through the definition of a shared semantic model — a digital standard that facilitates consistent understanding, integration, and communication across the port terminal sector.
Gaps	Identified standardization gaps	None.
Application areas		
Assets		X
Communication		X
Business		X
IoT		X

Table 37. Standard NgGSI-LD API

Applied standard	ID and Title	ETSI GS CIM 009 – **NGSI-LD API** standard
	Date of publication	2020
	Publisher	ETSI (European Telecommunications Standards Institute)
	License	Public
Pilot mapping	aerOS pilot	**Pilot 5**
	Technology name	Context Broker (Orion-LD, developed by FIWARE Foundation); Edge and Cloud integration
	Usage specifics	Orion-LD was used to enable semantic interoperability and linked data modeling for sensor data and metadata management across systems.
Gaps	Identified standardization gaps	
Application areas (Mark if applicable with "x")		
Integration		X
Communication		X
Information		X
Artificial intelligence		X
IoT		X

5. GDPR Compliance

This section analyses the GDPR compliance of the aerOS pilots. As presented in deliverable D2.5, a checklist has been created and hosted on the project servers. This checklist aims first to define the profile of the pilot according to how they manage personal data. Three roles have been defined:

- **Data controller:** This role is responsible for deciding why and how personal data should be processed within the pilot.
- **Data processor:** This role is responsible for processing personal data only on behalf of the data controller.
- **Data subject:** This role has its data stored or processed.

Depending on the processing of the data in the pilot, the roles are initially defined. Once these have been defined and chosen in the questionnaire, the questionnaire presents the questions relevant to the chosen roles.

The questionnaire analyses different aspects required by the GDPR. By means of these questions, the pilot responsible have selected the sections that comply by ticking on the questions. Once all the questions have been analyzed, a button at the end of the questionnaire has been used to download the results obtained so that they can then be analyzed and define the GDPR compliance of each of the aerOS pilots.

The Data Manager and Ethics Manager of the project (appointed at the beginning of the action) have supervised this activity to ensure compliance with aerOS best practices and the Data Management Plan.

The results of the questionnaires for each pilot of the project can be found in within the Project repository[25].

Pilot 1.1, Pilot 1.2, Pilot 1.3, **Pilot 2,** and **Pilot 4** have indicated that they do not make use of personal data in the pilot. For this reason, the questionnaires related to these pilots are empty.

Pilot 1.4 has indicated that they comply with all but two points of the survey: one of them is related to the pilot's operations outside Europe, and they have indicated that they do not have such operations; whereas the other point that they do not comply with relates to the performance of the Data Protection Impact Assessment (DPIA) for the processing of high-risk data, where they have indicated that they do not process sensitive data in the pilot. Therefore, it can be stated that Pilot 1.4 is GDPR compliant.

Pilot 3 indicated that they have the roles of data processor and data subject. In the controls associated with these roles, they have indicated that they comply with all controls. Therefore, it can be stated that Pilot 3 is GDPR compliant.

Pilot 5 have indicated that they only have a Data Controller role. They have also indicated that they comply with all the controls associated with this role. Therefore, it can be stated that Pilot 5 is GDPR compliant.

[25] Public URL to NextCloud repository: https://nextcloud.aeros-project.eu/s/EDeanYfwXLK56tS

6. Open Call Results

6.1. Lessons Learned from Open Calls

One of the tasks that has been performed in WP5 along the months M12-M38 has been the continuous support and observation of the execution of Open Call projects. aerOS reserved (as per Grant Agreement) a total of 900.000 € for Financial Support to Third Parties, which has materialized in 15 projects (60k€ each) aiming at applying aerOS technologies to specific challenges and enlarging our capacities to further verticals (not covered in the project).

The execution of such projects was structured in two rounds.

The first one (encompassing a total of 7 projects) was focused on individual projects targeting the provision of added value to aerOS pilots. In particular, their goal was to provide an innovative solution in the field of Cloud-Edge-IoT adjusted to one of the challenges that aerOS pilots defined. Requisites for their successful completion also included the delivery of packaged software components and the (virtual or physical) integration with aerOS Infrastructure Elements functioning in the pilots. The projects of this round concluded successfully in January 2025, and some of the conclusions obtained are reflected in this section. As a summary, all 7 of them were successful, delivering a solid demonstrator of innovative tools that provided added value to aerOS' pilots. Additionally, several cascade funding projects here contributed to aerOS in different ways, for instance providing datasets that were produced during their works.

The second round (encompassing a total of 8 projects) is still, by the time of concluding this deliverable, in its final stages (will finalize by the end of the current month – August 2025). However, it has already left a remarkable imprint in aerOS, having been able to improve the technological outcast of our technology. In addition, it has served as the ideal catapult to mobilize new verticals (than those already covered in the projects), demonstrating the scalability and uptalking potential of aerOS.

During the course of all projects in both rounds, a devoted team of aerOS (led by UPV and NCSRD) has been in charge of continuously supporting technologically and administratively their operations. Many interactions have taken place (including virtual meetings) which has helped both parts. On the one hand, Open Call projects have been able to install / modify / adopt aerOS technologies, being able to understand the inner functionality and to adapt it to their own goals. On the other hand, aerOS **Meta-OS has benefitted greatly** from this action since it has helped polish the material for installation or utilization. According to the Consortium, the conduction of Open Calls has meant a **huge leap in the portability easiness and adoption potential of the solution**. Beyond that, **it is being a crucial boost** for the upcoming Open-Source product **ECLIPSE stratOS**, which will already count with a solid baseline of beta testers.

This section is aimed at documenting the contributions from the Open Call projects to aerOS. Note that **this is not a thorough report on the activities done by the Open Calls**. Such projects had their own work plans and deliverables, that were duly submitted to the Project Coordinator (as specified in the Sub-Grant Agreement). However, these are not described nor summarized here, since that is not the goal of this deliverable. The results of the Open Calls (software, documentation, reports, etc.) are properly stored by the Project Coordinator, and may be available under specific request.

In the following pages, the report is focused in discussing whether the projects have met their goals, which aerOS technologies they have used, the level of integration of aerOS, which components have been utilized, what are the lessons learned, and whether or not any datasets have been shared.

Regarding the latter, some projects emanating from the 1st Open Call stepped in and provided datasets that were generated during their projects. These are of different nature, ranging from computing infrastructure monitoring up to training data used for ML models, or results of their experiments. These sets have not been included in this report, since they will be incorporated in the Zenodo repository of the project by M38 (October 2025).

Also, it is relevant to mention that, because of the unconcluded stage of the projects of Open Call #2, a new section will be incorporated in the next deliverable of the WP (D5.6, due to M38 – October 2025) where the final conclusions out of the 8 projects of OC#2 will be reported.

6.1.1. Open Calls #1:

The first round of Open Calls was focused on external 60k€ projects tackling specific challenges defined by aerOS pilots. In order to properly illustrate how such projects contributed to aerOS, it is compulsory to recap on the challenges defined by the stakeholders. Within Appendix A, it can be observed those challenges defined together with the final list of awarded projects. As it can be seen, the final distribution among challenges was uneven, having more projects devoted to P1.1 (SIPBB) and P4 (Port of Limassol) than to the others. This is not considered a drawback but a gauging of global interest.

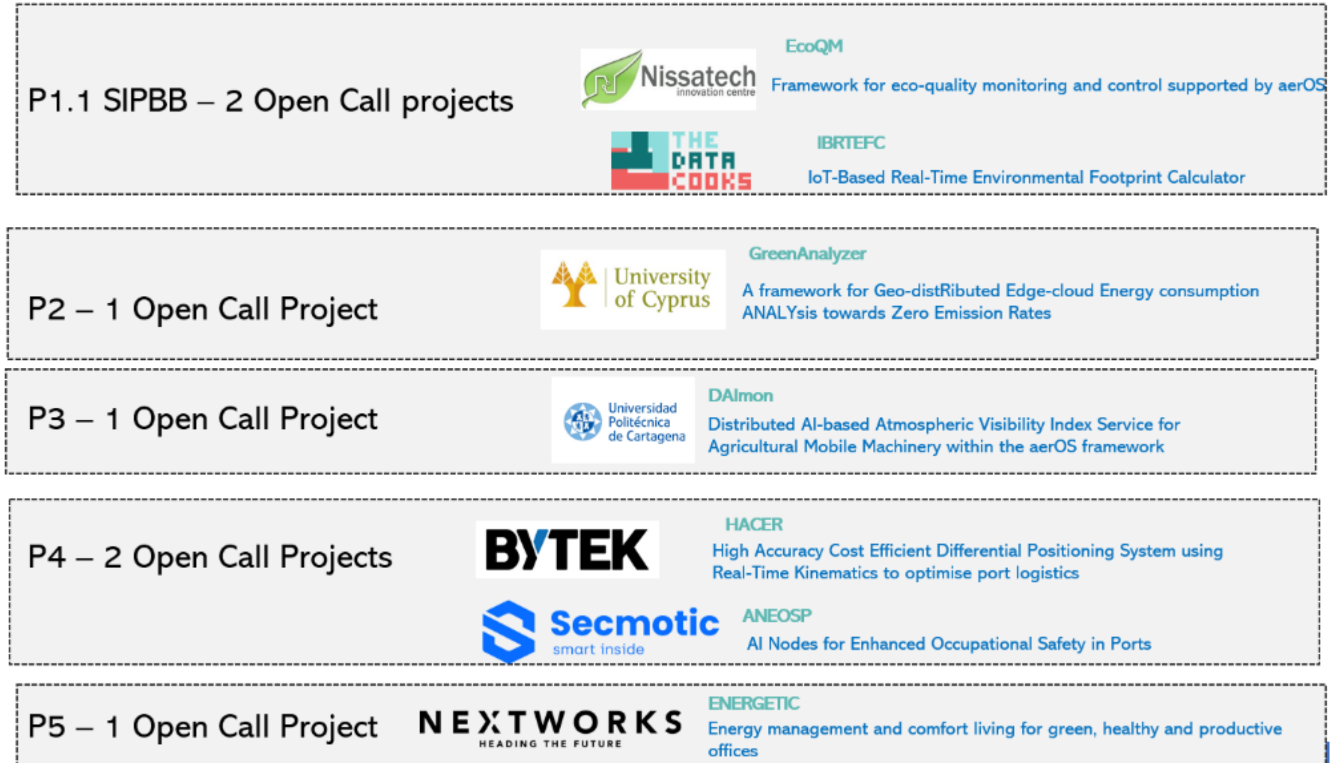


Figure 266. Pilots and their Open Call 1 Winners

After the finalization of the project, an evaluation was performed. The goal of this evaluation was double-edged: first, to confirm that the projects have achieved their goals; and second, to analyze the level of contribution to the project, including the integration of technologies and the added value provided in different forms.

Such evaluation was conducted in January 2025, deriving in the following analysis. The table below should serve to justify the level of cohesion of the Open Call projects with aerOS technology, as well as to settle the core material evidencing the mutual learning performed by both parts.

Table 38. Analysis of value provided by Open Call #1 projects

	ENERGETIC	GreenAnalyzer	DAImon	ANEOSP	HACER	EcoQM	IBRTEFC
Substantial results	A Telegram chatbot, integration, Dynamo system, chatbot functionality.	An energy profiling, stressor, datasets of energy (that have been shared). Integration of external APIs was also achieved. FedBed framework was also developed and delivered. Images updated to DockerHub.	Sensors and camera installation. AVI algorithm creation, development and deployment. Also, frugal Ai and training of such model in a distributed manner (AI task manager developed). Also, a UI was provided.	IoT-edge nodes created by SECMOTIC. Software for monitoring, notifications and user interfaces. Also, a successful trial took place. 5 models out of camera images.	Rover development, including firmware and hardware. Development of RTK algorithm and sending accurate GPS data via MQTT.	Two use cases - Manufacturing SME and machinery in NIS - well done and explained in the report - which is based on aerOS Federated Broker and data observation service. Second one is aerOS validation in Pilot 1 - SIPBB. The exchange and integration with the pilot has taken place.	DPP product based on their carbon footprinting tool BILEN, and also integrated with areOS and using data from SIPBB. Environmental Footprint Calculation System.
Demonstrating usage of aerOS components	Yes, integration with aerOS Data Fabric installed in OTE.	Yes. Federated Learning workloads in Pilot 2; deployment of FL services via aerOS Portal and HLO….	Yes, installation of aerOS, creation of domains, and integration of AI task manager with aerOS, together with device management code (IEs of aerOS). Also, deployment of AVI service via aerOS HLO.	Yes. Integration of their components within aerOS infrastructure. Data through aerOS Context Broker…	Yes, installation of self-awareness and self-orcehstrator in the rovers. Also, communication with Context Broker (good domain strcuture) was achieved.	Yes, in both cases.	Yes, self-awareness, self-orchestration, Data Fabric (via API).
Provision of datasets	**Yes.**	**Yes, large in size and great contribution.**	**Yes, sensors and Images from camera. Large size in Nextcloud.**	**Yes, link within deliverable D4.**	**To be provided under the whitepaper to be submitted later. Raw data provided to Project Coordinator.**	**Yes, submitted to Project Coordinators.**	**Data raw from SIPBB generated for this Open Calld.**
Level of on-site (or virtual) integration with pilot data and equipment	Yes - 60 sensors of 16 types across 5 rooms in OTE building.	Yes, including capturing of metrics about equipment via self-awareness.	Interaction with P3 stakeholders was based on validation of concept and interest of experimentation fields.	Yes, on-site visit and demonstration, including enough metrics (performance, AI model efficiency…).	Yes, deploymnent of the rovers (IEs) in the Port of Limassol. Result was successful.	Yes - environment data from SIPBB, and machinery data from Nis manufactring case. Also, computing-related information from self-awareness is monitored and integration is demonstrated.	Yes. Data about drone production line. Connection with SIPBB0s API and created other demos for commercial purposes.
Provide added value to a pilot	Yes, to pilot 5. Evidences of collaboration were clear.	Many nice insights and conclusions about usage of RES, CO2 footprint, etc.	Frugal AI application to the developments related to the pilot.	. To enhance port security with local processing and empowered CV models that do not overlap with the current brochure of aerOS.	Yes, GPS accurate positioning service made available to pilot partners.	Yes, provisioning of a packaging for simulating a manufacturing case.	Yes, a Digital Product Passport available from the Open Caller.
Show a demonstrator reflecting the results	Yes, they showed the data integrated. Also, a demo of ORCO chatbot	Online	Asked to show the UI to select the input data, analyze the pictures and obtain numerical results. This was presented in the previous video.	Yes. A demonstrator made, that includes data from the port of Limassol, thus integration is endorsed.	Online.	Yes. Data explorer of data coming from self-awareness.: Also the alarms generated out of the CPU usage, etc.	Flow described and demo performed, very satisfactory.
Objectives achieved	Yes	YES. The commitment about RES in the pilot was not possible to be done (P2 does not have); however they provided it from their PV pannels.	Yes, included TRL reaching.	Yes	Yes	Was explicitly asked.	Yes
License / open source strategy	Yes, Open Source code..	Yes, and already published: https://ucy-linc-lab.github.io/GreenAnalyzerProject/	Code provided to aerOS. Will follow an OSS Strategy.	Yes, Open Source code..	Yes, Open Source code..	Yes, Open Source code..	Yes, and already published in GitHub.
Further exploitation of the results	. Proper justification after question in the technical check. To be itnegrated in their commercial provision.	Demonstrated carbon emission reduction capacity of aerOS together with RES.. Great exploitation in terms of open source, academy and collaborative opportunity. UPSCALE is guaranteed because they will apply aerOS and the methods of GreenAnalyzer in the to-be-installed 86000 tones PV park in U. Cyprus.	Some aspects mentioned: partnership with local farmers and Smart Cities.	Participation in the incubation programme: Gran Canaria and La Lonja de la Innovación (Huelva).	Mentions in the report.	Yes, a solid business model was presented in the technical check.	Commercial feedback received from the market is very positive. Also, a go-to-marlet model was delivered. Also, **one research project accepted out of the results**..
Scientific dissemination / publications	Participated in a conference last June. Used the EU emblem and logo. Included in the first report.	2 papers in conference: IEEE IC2E and IEEE/ACM UCC. Website, GitHub repo, datasets and mention in MSc course. Another one to be submitted this month (to a conference workshop).	Yes, mentioned properly in the report and in the presentation.	No, only at commercial level. This is OK, since this was not a commitment from their proposal.	A whitepaper was published.	Yes, DEBS2025 and BPM 2025. Also a journal paper. Confirmed that all activities have followed the acknowledgment.	Video from FAKUMA Exhibition was provided to aerOS Open Calls coordinator. Promotion with Whirlpool in EIT Manufacturing

In addition to the previous gathered information, the companies awarded with an Open Call project in the first round were requested to fill a specific document to gather feedback about aerOS. In particular, the following template was utilized to gauge the level of maturity in the adoption readiness of aerOS technology. The responses from the Open Call awardees are included in Appendix B.

Question	Answer
How easy is aerOS to be installed and managed?	
Is aerOS ready to be deployed in a functional environment?	
Would you recommend adopting aerOS technology?	
Is aerOS aligned with your current strategy and frameworks?	

Please, list the most important issues that you encountered when using/installing aerOS technologies

Issue description	Resolution	Additional comments

Figure 267. Template of questions & answers, and feedback from OC #1 winners

6.1.2. Open Calls #2:

With regards to the second round of Open Calls, as it has been said it is not completed yet. Therefore, it is not possible to provide a table similar to Table 22 at this moment. However, it will be generated whenever the final check of goals completion will take place

After the proper selection procedure, a total of 8 projects were funded responding to the following data (that was properly communicated to the CSA, the EC, and via the aerOS official website), as it can be seen in the following table.

Table 39. List of projects of Open Call #2

#	Acronym	Company	Title	Country	Entity Type
139	SensorsReport on aerOS	SYNCHRO S.R.L.	SensorsReport.com On AerOS	Romania	SME
143	SecCon	ExcID	Transparency technologies for a secure continuum	Greece	SME
147	SecureOrch	Virtual Angle	Security Orchestration for Critical Infrastructure through aerOS	Netherlands	SME
150	FireGuard	OneSource	Forest Intelligence, Response, and Environment Governance for Unified Alert and Risk Detection	Portugal	SME
152	MOTION++	UniWA	iMage and lidar fusiOn for vehicle exTerior InspectiON ++	Greece	University
154	safeOS	Plegma Labs	A self-healing AI solution for real-time safety monitoring using a Meta-OS paradigm	Greece	SME
156	SPARTA	TU Shannon	SPort Analytic Toolkit for Real-Time AeROS Applications	Ireland	University
164	AIRCATT	Internet of Thinger	Artificial Intelligence-based early detection of railway cracks by combining aerOS and Thinger.io technologies	Spain	SME

Nonetheless, several lessons have been already learned from the execution of these projects. As a matter of fact, given the maturity level of aerOS solutions (WP3 and WP4 already finished, and integration in pilots at its final stages), the focus of interaction has been in troubleshooting whenever facing the installation / customization of aerOS technologies in the premises of the new verticals proposed by the OC#2 projects.

- **New features in the management portal**:

  - Retrieval of a Docker image / Helm chart from a private repository instead from the public Docker Hub / Artifacts repository, indicating and encapsulating the credentials.
  - Increase the available customization options in the Management Portal's Helm chart
- **Adjustments in the installation and automation**:
  - The pre-requirements (e.g., existence of K8s cluster) have now a more prominent space in the installation guide.
  - Improvement of the configuration of the Identity Management component (KeyCloak) in the configuration of needed mappings with LDAP.
  - Simplification of procedures in the interim between main domain and other domains.
  - HLO versioning and redpanda topics are now better depicted in the guide.
  - Automation of the Organization (Domain owner) entity creation in the Federator
  - **Networking issues**:
  - Optimization of the values to be indicated related to aeros-k8s-shim utility (intended on simplifying the connection with IdM).
- **Logging and debugging functionalities**:
  - Added more details in self-orchestration log messages and API prompt logs.
  - HLO and redpanda topics are now better communicated and the messages are clearer.
- **Packaging and validation aspects:**
  - Validation of the developed Helm charts for the LLO: LLO API, LLO K8s and LLO Docker
  - Validation and testing of the LLO Docker
  - The aeros-k8s-shim has been moved to an independent chart to facilitate the installation process because it was previously included inside the global HLO chart. This new chart has been validated in the Open Calls of OC#2.

7. Conclusions

This deliverable has demonstrated that aerOS is now fully deployed and operational across all pilots, with integration achieved in realistic industrial, energy, mobility, port, and building environments. The work completed between M19 and M35 confirms that the Meta-OS can support heterogeneous infrastructures, manage workloads across the Edge Cloud continuum, and interconnect domains through federation and standards-based interoperability.

The pilots showed tangible improvements in their respective domains, streamlined metrology and defect detection in manufacturing, flexible orchestration of production assets, real-time monitoring of energy and environmental data, predictive maintenance in ports, and smart-building optimization enhanced with 5G capabilities. In each case, aerOS contributed to higher efficiency, better responsiveness, and more reliable system operation, while respecting security and compliance requirements.

The OCs complemented these results by bringing in external adopters who tested the platform under new conditions. Their feedback directly improved usability, documentation, and packaging, helping to reduce entry barriers and increase the maturity of the platform for a wider ecosystem.

Overall, the activities reported in D5.4 confirm that aerOS has moved from early deployment into stable operation across complex and diverse pilots. The results provide a robust basis for the next phase, where final validation activities and KPI assessment in D5.6 will measure the impact and performance of the platform at scale, ensuring that the lessons learned translate into actionable improvements and potential long-term impacts.

Appendix A – Pilot Challenges for Open Calls #1

Table 40. Pilot 1 Challenges Open Call #1

Code	Name	Description
P1C1	Real time footprint monitoring in production processes	Implement a system that can measure and communicate the environmental footprint of production processes in real time. The environmental footprint is a comprehensive indicator that covers 16 environmental impacts, such as climate change, water use, resource depletion, and toxicity. The system should be able to collect and analyse data from various sources, such as sensors and other equipment operation parameters, along the manufacturing process (e.g Product specific data, process machine data such as temperature, humidity, energy consumption...) . The system should also be able to provide reliable, verifiable, and comparable information to the stakeholders, such as manufacturers, consumers, regulators, and investors by feeding a preliminary DPP (Digital Product Passport). The system should also be able to support decision making and optimization of the production paths to reduce the environmental footprint and improve the sustainability performance.
P1C2	Optimization of AGV paths	To find the best way to plan the routes of automated guided vehicles (AGVs) in logistic processes by facing some difficulties, such as avoiding collisions, adapting to dynamic environments, and minimizing the travel distance and time. The challenge requires optimizing the AGV paths in logistic processes, taking into account various factors and constraints, such as the layout of the environment, the location and demand of the tasks, the number and capacity of the AGVs, the traffic rules and safety regulations, and the energy consumption and maintenance costs, in order to avoid saturation in the assembly line, outsourcing to other production lines considering environmental aspects (e.g most efficient & carbon neutral path).
P1C3	Remote operation of CMM (Coordinate Measurement Machines)	To enable the remote operation of coordinate measurement machines (CMMs) that can measure the geometry and quality of physical objects with high accuracy and precision. CMMs are devices that use probes or sensors to sense discrete points on the surface of the object and display them in a digital format. The challenge requires a low latency control and monitor the CMMs from a distance, using a computer or a mobile device. The system should also provide real-time feedback and data visualization to the users, as well as enable the communication and collaboration among different stakeholders. The system should also ensure the security and reliability of the data transmission and storage.
P1C4	Managing industrial production applications with Behaviour Trees	The solution sought should be capable of controlling complex algorithms, particularly through behaviour trees or other Low Code Tools, which are essential for managing complex industrial production flows with flexible automation assets. Using such tools should not necessitate extensive coding or deep programming knowledge, instead providing a visual interface for managing, modifying, and executing these algorithms. This effectively reduces the dependence on manual coding. Additionally, the solution should allow users to visually construct and alter decision paths in low code tools such as behaviour trees using straightforward features such as drag-and-drop. The flexible, adaptable solution needs to be user-friendly and accessible to users ranging from beginners to experts, ensuring that the accuracy and functionality of production flows are not compromised.
P1C5	AGV travels optimization	To train an AI/ML algorithm there is the need to generate a synthetic dataset of simulated product orders over time (integer numbers) arriving from simulated companies outside. The dataset should be able to highlight some possible faults/inefficiency of the automatic factory, for example bringing saturation of the line or alternating high demand periods with low demand periods and let the system solve/improve/optimize the AGV travels management system. This challenge would require the creation of such data set. Then, drawing from these data, the challenge would appreciate the creation of an automatic real-time random order generator that simulates the orders arriving from simulated companies outside and a data monitoring dashboard that shows these data on the screen.
P1C6	Designing Plug-and-Produce Factory Modules for Agile Production Environments	The challenge is to design and develop a generic, CE compliant and modular production asset that can be easily integrated into a factory as plug and produce factory modules. These assets should be designed to be flexible and adaptable, allowing for frequent changes in their mission and position within the factory. The challenge is to create a system that enables easy and seamless integration of these modules into the factory environment, including the communication via OPC UA and/or ROS2, without compromising efficiency or productivity. The modules should be designed to be easily interchangeable, allowing for quick reconfiguration of the factory layout to accommodate changing production needs. The ultimate goal is to create a highly agile and responsive factory environment that can rapidly adapt to changing market demands and production requirements. Examples: Handling or Quality Inspection machineries.
P1C#	Global	Others fitting within the global challenges descriptors (see A.2 -

Table 41. Pilot 2 Challenges Open Call #1

Code	Name	Description
P2C1	Advanced context-specific energy level prediction on the edge	The motivation for this challenge is provided by the following scenario. Consider a computing solution that uses UPS (Uninterruptible Power Supply) to avoid full discharge of an element carrying out some tasks when we do not have main power source from renewable energy source. Here, tasks processing should be interrupted before a certain UPS battery level and delegation of tasks should be suspended after determining the appropriate battery level. When the power is back the system should restore the tasking and usage of the edge computing solution. Here, an elaborated algorithm is needed that should take into account UPS battery level, time, weather forecast, size of workload and any other use-case/environment specific parameters that may help in predicting the energy level at the edge. Core of the challenge is to boost the accuracy of prediction of energy consumption of hardware items during workload execution in order to estimate time to shutdown, that will in turn be used by aerOS orchestration algorithms. This challenge can be further extended by taking into account temperature of the container, servers and predict time to over/under heating that will hinder functioning of an element.
P2C2	Carbon intensity prediction	To limit aerOS environmental impact and contribute to the goals of European Green Deal a conscious energy consumption is required. Smart management and scheduling tools of aerOS call for information on carbon intensity of available energy across Europe. aerOS, in pilot P2, could benefit from solid, actionable current and predicted emission data to optimize workload distribution and processing both spatially and temporally. Essentially perform energy-intensive activities in place and time that offers the energy with lowest carbon intensity. In the scope of this challenge, a prediction algorithm is expected that combines publicly available data to produce a map of carbon intensity across Europe, with special emphasis in the Poland region (where the pilot takes place). Spatial and temporal resolution will be deciding factors in the evaluation process.
P2C#	Global	Others fitting within the global challenges descriptors (see **A.2** -

Table 42. Pilot 3 Challenges Open Call #1

Code	Name	Description
P3C1	Low-cost dust detection system	During secondary tillage, a large amount of dust can be generated if the soil is dry, blocking the view of the ground. Currently, a LIDAR is used to detect the dust, but it is expensive. To guarantee a reliable use of cameras analyzing the ground, a more cost-efficient system or methodology [both hardware and software solutions are possible] is needed to enable wide adaptation. This should detect the dust and provide a metric representing the degree of current visibility restriction or the quality of current possible ground detection.
P3C2	Seamless wireless communication in challenged agricultural environments	In many agricultural fields, coverage by terrestrial radio networks is often insufficient. However, digital agriculture requires connectivity to the Internet as well as between the machines in use, to enable effective processing of tasks in the fields. In order to further put technological advancements in digital agriculture into practice, a system is needed that enables seamless, uninterrupted, low-latency communication to the Internet in these challenged fields as well as when transitioning to such fields.
P3C3	Cost effective, short-range sensor fusion for full area coverage around moving vehicles/machinery	Work processes become more and more autonomous so reliable object detection and classification in the proximity of moving vehicles/machinery is essential. For moving vehicles/machinery it is essential to have full coverage of short-range area surveillance around the vehicle. In particular to the rear, often machine parts block vision of certain sensors and create blind-spots. Additional complexity is added by harsh-environmental, out-door conditions, which shall not lead to degradation of the performance (fog, rain, snow, uneven terrain, etc.). Also the machine itself may add unfavourable effects, such as vibration, heat, etc. The existence of high bandwidth connections of the vehicle cannot be guaranteed, which leads to an embedded solution. A solution by combining a diverse set of sensors and fusing the data may be the obvious approach. The analysis shall cover the performance requirements on the computing and data bandwidth to be mastered in the embedded solution and a qualification of advantages, disadvantages of the chosen sensor combinations in several aspects: performance of object detection/classification, cost and installation complexity, etc.
P3C#	Global	Others fitting within the global challenges descriptors (see **A.2** -

Table 43. Pilot 4 Challenges Open Call #1

Code	Name	Description
P4C1	Low-cost accurate GPS	Centimetre accuracy D-GPS is an interesting feature in the port environment to monitor the location of assets, but hardware is too expensive. Using normal GPS receivers and performing a post-processing of the signal can alleviate the costs and keep almost same location accuracy. The goal of this challenge is to propose and implement a (new hardware) PoC of a location system providing real time location with a precision of a few centimetres.
P4C2	Secure private mobile network	A port container terminal is a relatively large outdoors area. In general, the connectivity of the different assets is provided by the installation of fiber optics, which leads to very high deployment costs. The goal of this challenge is to test a wireless network infrastructure that can guarantee enough coverage for the future connected straddle carriers (whose data will be collected and transmitted) along the pilot area under test. Moreover, since the sensitive information to be transmitted, regular public 4G/5G networks are seen as direct candidates for malicious attacks. Thus, the challenge aims at the deployment and test of a private 4G/5G network deployment that can guarantee privacy and security in the transfer of telemetry data acquired in all port assets. Innovation must be clearly posed by going beyond current commercial approaches that have proven insufficient in such environments.
P4C3	New predictive maintenance or computer vision use cases	Project partners predictive maintenance use cases are focused on trolley wire rope enlargement and load cell measurements. New predictive maintenance use cases (new AI services), such as tyre pressure monitoring are of interest. On the other hand, pilot is already developing computer vision AI models for detecting deformations in containers, but the live video feeds can be used to develop models for improving workers' safety, like detecting Straddle Carriers or other yard equipment below an STS crane under operation (these cases are not under the current scope of the pilot).
P4C4	3D digital twin visualization	Pilot partners already provide 2D visualization of some port assets (cranes, buildings, container blocks, etc.) taking as baseline an AutoCAD model of the terminal. In order to provide a more close to real life environment, this challenge seeks extending 2D visualization to a 3D environment using real-time location information of assets and asset status data visualization (mostly using telemetry data and container stock information). Here, connection to aerOS is relevant in terms of being able to dynamically plot and visualize values related to the objects represented in the 3D model.
P4C#	Global	Others fitting within the global challenges descriptors (see **A.2** -

Table 44. Pilot 5 Challenges Open Call #1

Code	Name	Description
P5C1	5G SA IOT Gateway	The pilot would benefit from the incorporation of 5G SA IoT gateways. Current commercial versions do not provide open gateways that could be used for installing aerOS functionalities. In addition, it would be needed that such gateways would work in a 5G standalone fashion and would operate inside a Smart Building (several of them would be appreciated). In addition, the gateway should include capabilities of SDN, NFV (virtual network functions) and Network Applications. Innovation is sought in this challenge with regards to plug & play nature for an agile deployment in the Pilot 5 of aerOS.
P5C2	Energy Consumption Calculation	The goal of this challenge is to found an energy consumption calculation system for a sensorised building. COSMOTE Lab building incorporate a series of sensors across various floors (more info in D2.2). The challenge would be to deliver a smart algorithm to calculate the energy consumption of the activities taking place in the building, alongside a system for gathering such metrics and exposing them to the aerOS meta Operating System. Accurate measurements and predictions would be required in order to fine-tune the actions that will be actioned upon aerOS innovations.
P1C#	Global	Others fitting within the global challenges descriptors (see **A.2** -

Appendix B – Q&A and Feedback from Open Calls #1

The seven projects of OC#1 were asked to fulfil the abovementioned questionnaire. The goal behind this exercise was to understand the main drawbacks of the technology (and, mostly, its documentation) from aerOS. The lessons learned mentioned below were used during the period M29-M35 to improve our positioning towards the community. The comments below must then be understood as an informal evaluation of an incomplete product that has now been advanced.

ENERGETIC

Table 45. OC#1 – ENERGETIC feedback

Question	Asnwer
How easy is aerOS to be installed and managed?	The installation and management of aerOS is well-structured and modular, mostly thanks to the usage of cutting edge, open source and widely used containerization tools like kubernetes, which simplify the deployment and integration process
Is aerOS ready to be deployed in a functional environment?	aerOS is ready to be deployed in cloud oriented environments, where virtualization technologies allow the possibility to install virtualized componments packaged as HELM Charts
Would you recommend adopting aerOS technology?	Yes, adopting aerOS technology is reommended for organization seeking a unified platform to build virtualized platform in a standardized and reproducible way, focused on interoperability, security and efficient orchestration.
Is aerOS aligned with your current strategy and frameworks?	Nextworks' commercial product (Symphony) is built as a set of containerized software components running in Kubernetes. Therefore, the aerOS vision is aligned with the company strategies

GreenAnalyzer

Table 46. OC#1 – GreenAnalyzer feedback

Question	Asnwer
How easy is aerOS to be installed and managed?	The primary challenge arises from aerOS services that require a Kubernetes (K8S) cluster to run, which is a limitation of our current capabilities rather than the aerOS project itself. To test a fully functional deployment, we utilized the Pilot 2 infrastructure, which provided a pre-installed aerOS setup.
Is aerOS ready to be deployed in a functional environment?	If the environment has a Kubernetes cluster installed, aerOS can indeed be used for service orchestration and observability.
Would you recommend adopting aerOS technology?	Yes, especially for users who are not familiar with Kubernetes low-level commands.
Is aerOS aligned with your current strategy and frameworks?	Yes

Issue description	Resolution
Installation in a non-Kubernetes environment	Some services were still note ready for Docker alone. This is being addressed by the Consortium. NCSRD provided with access to specific services within a deployment. Later, the Pilot infrastructure granted us access to its deployment, enabling us to run our experiments.
Varying levels of maturity in the documentation	Overall, the documentation is in good shape, but there are some minor inconsistencies. For example, some components have overly detailed descriptions, while others are too high-level. The consortium could consider harmonizing the level of detail across the documentation.
The UI is not fully integrated with the underlying system	While the aerOS UI is intuitive and user-friendly, certain functionalities, such as editing an already created application instance, are missing. Although users can access the desired information and functionality directly through the APIs of the underlying components, these features should also be made available through the aerOS UI.
Visibility into the underlying system	For a user to have full observability and logging of their application, they must communicate directly via Kubernetes or the APIs of the aerOS components.
Restricted number of services	A minor issue with the system is that the UI restricts the deployment of more than 10 services on the infrastructure, while the underlying system appears to support this through its APIs.

DAIMon

Table 47. OC#1 – DAImon feedback

Question	Asnwer
How easy is aerOS to be installed and managed?	The installation of aerOS has been straightforward, although the management is a bit more complex. This is because there are many applications, and connecting them all to work properly can become somewhat challenging.
Is aerOS ready to be deployed in a functional environment?	We believe that aerOS could be deployed in a functional environment, as it meets all the requirements for a solid and scalable infrastructure in large projects. It includes layers for user management, authentication, and encryption to work with a secure and scalable structure.
Would you recommend adopting aerOS technology?	Yes, we would recommend it. It provides a solid foundation to work with, and it includes very reliable and widely used technologies that can be extremely helpful. For example, it facilitates the implementation of artificial intelligence techniques such as federated learning.
Is aerOS aligned with your current strategy and frameworks?	aerOS is an excellent choice since the focus in our project is to apply distributed artificial intelligence techniques. aerOS aligns well with the needs of our work, providing a secure and functional framework for the devices we deploy. We also believe that it will provision the features to support the growth of our solution for agricultural machinery, helping us to escalate it without the need of constant modifications to manage newly deployed infrastructure elements.

Please, list the most important issues that you encountered when using/installing aerOS technologies

Issue description	Resolution
We had doubts about which Domain ID we should use.	To solve this problem, we contacted the team responsible for technical support during the installation, and they told us we could choose any Domain ID we wanted. We believe it would be very helpful to include this information in the guide.
When installing, we were not sure if we should create multiple namespaces.	When we were installing the applications, we had a doubt about whether we should use different namespaces in Kubernetes to separate the central node from the edge devices. We decided to do so because, after analyzing the aerOS infrastructure, we believed it was the most reasonable approach.
User management for LDAP and Keycloak	The guide provides a structure for organizing users in projects, although it can be overly complex, especially for small-scale projects. Additionally, in the PowerPoint presentation that outlines the steps to follow, the password is omitted, requiring you to search for it in order to proceed with the next steps.

ANEOSP

Table 48. OC#1 – ANEOSP feedback

Question	Asnwer
How easy is aerOS to be installed and managed?	Deep technical knowledge is needed. A junior / mid developer team could had issues tackling the installation. The senior team had troubles finding the correct way, although there was great support. The installation difficulty is medium for a highly qualified tech team
Is aerOS ready to be deployed in a functional environment?	Yes it is. aerOS leverages on very well tested technologies to add the functionality layer desired
Would you recommend adopting aerOS technology?	It will very much depend on the use case. I would recommend adopting the technology when things are a little bit easier to deploy / understand / manage
Is aerOS aligned with your current strategy and frameworks?	aerOS is almost fully aligned with our strategy and frameworks. There is at the moment a dependency on Kubernetes, but when Docker is fully supported, it will be fully aligned

Issue description	Resolution
Documentation. When using the technology, aerOS claimed to be fully operational in Kubernetes and Docker. At the 6th step of the installation, docker documentation was missing	After contacting support, they guided us and recommended to migrate to kuberetes
Docker & Kubernetes. Our platform infrastructure was based on a technology not ready to be used (Docker), so we had to create a specific kubernetes cluster instead of using docker in our development environment	Dedicated nodes for kubernetes cluster and deployment on separate instances
Self Modules hardware dependant. When using the self modules on platforms like NVIDIA Orin, these modules were prepared to use a certain type of architecture and it was using cpufreq_utils command, which was not available in this platform.	Pivot to use only Raspberry Pi instead of NVIDIA family (jetson Nano, Orin..)
Self modules do not work completely out of the shell. When using it in raspberry, some of the parameters were not read. We had to deep dive onto the python script inside the docker image of the SelfModules, do some modifications, and override that script (changing the out of the shell working objective of this module)	Logic overriding of the Self* modules, concretely the script.py
Logic for the InfrastructureElements in Kubernetes. Since we had to create a dedicate infrastrucutre with 4 medium machines in AWS, we set the cluster to be powered off every day at 20:00 and start at 8:00. This creates new infrastructure elements in Orion-LD and leaves some unused entities, and growing the database	Assume that this way of working was not expected by design. Create a script to clean the unused entities (nodes in kubernetes)

HACER

Table 49. OC#1 – HACER feedback

Question	Answer
How easy is aerOS to be installed and managed?	The installation guide feels incomplete, lacking explanations of what is being done in each step. Improvable content on how to create an IE and how self-* modules work. Assistance from the aerOS technical team has been required.
Is aerOS ready to be deployed in a functional environment?	It is not yet in a high TRL, withc some difficulties to adapt for new implementations.
Would you recommend adopting aerOS technology?	We are aware that this is a pilot and that all feedback will serve to improve in the future and make it more accessible.
Is aerOS aligned with your current strategy and frameworks?	The computational resources required for the full deployment of aerOS makes it unattractive for a resource constraint and embedded systems environment.

Please, list the most important issues that you encountered when using/installing aerOS technologies

Issue description	Resolution
Lack of detailed information about installation and implementation	Emails with technical questions and help.
Requirements of static and public IPs for each device to be implemented in the pilot.	Contact with SIM providers which could get us static and public IPs. The requirement is not in our hand, but in an extern provider

IBRTEFC

Table 50. OC#1 – IBRTEFCS feedback

Question	Answer
How easy is aerOS to be installed and managed?	Initially, we faced some challenges installing the aerOS tools as we had only set up two tools for the bare minimum configuration. However, we later realized an additional tool was needed to connect these components. After consulting with our mentor, we received further guidance and proceeded with the installation. During this time, the aerOS team also released an updated installation guide, which significantly streamlined the process. With this guidance and the improved instructions, we were able to successfully install and manage the aerOS tools with ease. We believe current level of the solution and documentation enable smooth installation.
Is aerOS ready to be deployed in a functional environment?	For initial deployments and testing purposes, it serves as a solid starting point.
Would you recommend adopting aerOS technology?	Yes
Is aerOS aligned with your current strategy and frameworks?	Yes, we believe aerOS is aligned with our containerization approach and effectively supports our current strategy and frameworks, particularly for monitoring resources. In terms of busines model, it fits with our strategy. We provide our solution to system integrators and they integrate our solution to industrial systems they provide. In that sense cloud-edge continuum can optimize their connectivity architecture and reduce costs.

Please, list the most important issues that you encountered when using/installing aerOS technologies

Issue description	Resolution
Pod Errors and Restarting Issues: Some pods, like self-awareness-powerconsumptionamd64, encountered repeated restarts due to configuration errors	After ensuring required images were pulled and correcting configuration parameters, the errors reduced but not fully resolved
Connectivity Issue: The Orion-LD Context Broker endpoint returned a "Service Not Found"	Reviewed and updated Helm parameters (with creating entities) to match the correct subordinate endpoint and reconfigured the node IP.
Missing Entities: Errors occurred when attempting to access entities within OrionID.	Entities were not created due to installing minimal tools. Additional tools/modules were required to populate the environment. Coordination with the aerOS team was necessary for troubleshooting.
Self- Module Errors*: Modules like self-awareness-hardwareinfo and self-awareness-powerconsumptionamd64 produced errors during operation.	Updated image tags as recommended (hardwareinfo.image.tag=1.3.0, powerconsumptionamd64=1.1.0-legacy-entity-id) but errors persisted