D7.3 – IoT-NGIN Living Labs use cases intermediate results

WORKPACKAGE	WP7	**PROGRAMME IDENTIFIER**	H2020-ICT-2020-1
DOCUMENT	D7.3	**GRANT AGREEMENT ID**	957246
REVISION	V1.0	**START DATE OF THE PROJECT**	01/10/2020
DELIVERY DATE	10/05/2023	**DURATION**	3 YEARS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 957246

DISCLAIMER

ACKNOWLEDGEMENT

This document is a deliverable of IoT-NGIN project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 957246.

The opinions expressed in this document reflect only the author's view and in no way reflect the European Commission's opinions. The European Commission is not responsible for any use that may be made of the information it contains.

PROJECT ACRONYM	IoT-NGIN
PROJECT TITLE	Next Generation IoT as part of Next Generation Internet
CALL ID	H2020-ICT-2020-1
CALL NAME	Information and Communication Technologies
TOPIC	ICT-56-2020 - Next Generation Internet of Things
TYPE OF ACTION	Research and Innovation Action
COORDINATOR	Capgemini Technology Services (CAP)
PRINCIPAL CONTRACTORS	Atos Spain S.A. (ATOS), ERICSSON GmbH (EDD), ABB Oy (ABB), NETCOMPANY-INTRASOFT SA (INTRA), Engineering-Ingegneria Informatica SPA (ENG), Robert Bosch Espana Fabrica Aranjuez SA (BOSCHN), ASM Terni SpA (ASM), Forum Virium Helsinki (FVH), ENTERSOFT SA (OPT), eBOS Technologies Ltd (EBOS), Privanova SAS (PRI), Synelixis Solutions S.A. (SYN), CUMUCORE Oy (CMC), Emotion s.r.l. (EMOT), AALTO-Korkeakoulusaatio (AALTO), i2CAT Foundation (I2CAT), Rheinisch-Westfälische Technische Hochschule Aachen (RWTH), Sorbonne Université (SU)
WORKPACKAGE	WP7
DELIVERABLE TYPE	REPORT
DISSEMINATION LEVEL	PUBLIC
DELIVERABLE STATE	FINAL
CONTRACTUAL DATE OF DELIVERY	30/04/2023
ACTUAL DATE OF DELIVERY	10/05/2023
DOCUMENT TITLE	IoT-NGIN Living Labs use cases intermediate results
AUTHOR(S)	J. A. García Ontalba (BOSCH), M. Urias Martinez (BOSCH), N. Akmaikin (FVH), Josep Escrig (I2CAT), M. Catalan (I2CAT), M. Montagud (I2CAT), Y. Kortesniemi (AALTO), D. Lagutin (AALTO), J. Autiosalo (AALTO), Ilkka Lakaniemi (AALTO), T. Velivassaki (SYN), A. Voulkidis (SYN), C. Betzelos (SYN), A. Gonos (OPT), H. Kinnunen (ABB), F. Bellesini (EMOT), M. A. Bucarelli (ASM), F. Santori (ASM), A. Corsi (ENG)
REVIEWER(S)	H. Keinonen (AALTO), J. Escrig (I2CAT), R. Mosahebfard (I2CAT), M. Catalan (I2CAT)
ABSTRACT	SEE EXECUTIVE SUMMARY
HISTORY	SEE DOCUMENT HISTORY
KEYWORDS	IoT-NGIN, Intermediate results, Trial site, Living lab

Document History

Version	Date	Contributor(s)	Description
v0.1	03/12/2022	BOSCH	Table of contents
v0.2	09/01/2023	BOSCH	Sections 1 & 2 preliminary
V0.3	12/01/2023	BOSCH, FVH, AALTO, OPT, SYN, I2CAT, ASM, ENG, EMOT	Sections 3 & 4 preliminary
V0.4	02/04/2023	BOSCH, FVH, AALTO, OPT, SYN, I2CAT, ASM, ENG, EMOT	Final document preliminary; Ready for peer review
V0.4.1	11/04/2023	AALTO	Peer review
V0.4.2	18/04/2023	I2CAT	Peer review
V0.5	24/04/2023	BOSCH, FVH, AALTO, OPT, SYN, I2CAT, ASM, ENG, EMOT	Peer review comments resolved
V0.6	09/05/2023	BOSCH	Document finalization
V0.7	10/05/2023	CAP	Quality Check
V1.0	10/05/2023	BOSCH	Final version

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AGLV	Automated Guided Land Vehicles
AGV	Automated Guided Vehicles
API	Application Programming Interface
AR	Augmented Reality
BLE	Bluetooth Low Energy
CV	Computer Vision
DLT	Decentralised Ledger Technology
DMP	Data Management Plan
DoA	Description of Action
DR	Demand Response
DSO	Distribution System Operators
FAIR	Findability, Accessibility, Interoperability, Reusability (of Data)
FL	Federated Learning
GAN	Generative Adversarial Networks
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
HDV	Human Driven Vehicle
IDAC	IoT Device Access Control
IDD	IoT Device Discovery
IDI	IoT Device Indexing
IPR	Intellectual Property Rights
KPI	Key Performance Indicator
LAN	Local Area Network
LL	Living Lab
MAD	Malicious Attack Detector
ML	Machine Learning
MLaaS	Machine Learning as a Service
MTD	Moving Target Defences

OBD	On-Board Diagnostics
ODC	Open Data Commons
ORDP	Open Research Data Pilot
PLC	Programmable Logic Controller
PMU	Phaser Measurement Units
PPFL	Privacy-Preserving Federated Learning
PQA	Power Quality Analysers
QR	Quick Response
RFID	Radio Frequency Identification
RL	Reinforcement Learning
SMX	Smart Meter eXtension
SSIs	Self-Sovereign Identities
ST	Semantic Twin
TBD	To Be Defined
TMC	Traffic Message Channel
TSN	Time-Sensitive Networking
UAV	Unmanned Aerial Vehicle
UC	Use Case
UWB	Ultra-Wideband
VLP	Visual Light Positioning
VM	Virtual Machine
VPN	Virtual Private Network
WP	Work Package

Executive Summary

This deliverable includes updates about the trial site set-up and other progress made in the IoT-NGIN Living Labs:

- Human-Centred Twin Smart Cities Living Lab, Finland
- Smart Agriculture IoT Living Lab, Greece
- Industry 4.0 Use Cases Living Lab, Spain
- Industry 4.0 Use Cases Living Lab, Finland
- Smart Energy Grid Monitoring / Control Living Lab, Italy.

Specifically, the document reports updates about the used equipment, data collection, alignment with the required IoT-NGIN technologies and sequence diagrams for the 10 use cases (UC), validating the IoT-NGIN framework in the Living Labs. Moreover, intermediate results of the test and validation processes are reported for each UC, according to the test and verification plan presented in the previous version of this deliverable (D7.2). The intermediate results include functional testing of the IoT-NGIN components in the Living Labs in the context of the defined scenarios. So far, the results verify proper operation of IoT-NGIN in the Living Labs and the next step is towards evaluating the value brought by IoT-NGIN innovations in the piloted use cases.

The final results of the Living Lab validation of IoT-NGIN will be reported in D7.4 "IoT-NGIN Living Labs use cases Assessment and Replication guidelines", due at the end of September 2023. This report will present the insights gained from the pilot execution, and will identify lessons learnt and guidelines for replication in other environments or use cases.

1 Introduction

The trial site set-up was completed and documented in D7.2 [1]. The deliverable also included the updated Data Management Plan (DMP) specifying data to be collected as part of the in-site trials.

DMP is stable and will be used as a reference for data collected during the first stage of in-site trial. Collected data will be used to evaluate intermediate results.

Based on activities documented on D7.2, the sites are ready for the trials. Necessary equipment was provided, or use cases were adapted as part of D7.2 and no changes are expected during activities documented in D7.3. Intermediate results evaluation will be based on assumptions made in D7.2, and actual data collected during activities documented in D7.3. Discrepancies between expectations of D7.2 actual results will be addressed and any update from D7.2 can also be addressed.

The rest of the document is structured following the pattern set in D7.2 to ease comparison of in-site trial results with specifications set in D7.3 as follows. Section 2 describes the methodology followed to collect the necessary information from the different trial partners. Afterwards, Section 3 details the advances in the different LLs and their use cases, based on the different aspects of the followed methodology and presenting intermediate results. Section 4 highlights the replicability of the IoT-NGIN architecture across different application domains and use cases. Finally, Section 5 concludes the document and outlines next steps to follow.

Since the DMP is stable and no changes are expected, section 4 of D7.2 has no counterpart in this document.

2 Methodology and trial coordination

Trial sites will run mostly independent from each other. Besides, real operation in a changing environment can dictate changes into the use cases unforeseen and the use case definition and planning.

This section will address the changes in the use cases during the in-site trial, including, but not limited to: use case definition, trial site status, actual data collected, testing/evaluation procedures, etc. A questionnaire similar to the one used to gather the information of each use case for D7.2 will be used.

The rest of this section provides a description of the main sections in the mentioned questionnaire, which also correspond to the sub-sections of the description of each trial UC as defined in Section 3 of D7.2.

2.1 Trial site description or update

This section is meant to be a short summary of the real site, specifying use-case operation and, where they happen, differences from provisions set up in D7.2

2.2 Used equipment

This section will list the used equipment for each trial site. If new equipment has been set in operation to address use-case needs, it will be included here with the reasons for inclusion and a detailed description matching those included in D7.2.

Also, if equipment procured during the activities covered in D7.2 could not be used, it will be specified here, together with reasons for not using it and the time span it was in use.

2.3 Data collection

This section covers the description of actual data collected during the trials. This will include data amount, time span of data collection and, when needed, differences with datasets defined in the DMP, as well as provisions taken to match actual datasets to planned ones.

2.4 Alignment with IoT-NGIN technologies update

This section will document any change from the alignment documented in the corresponding section of D7.2 For each change, a detailed description will be included, stating reason for change, actual change contents, effects of change in the intermediate results and potential long-term effects for the trial site results.

2.5 Use case sequence diagram(s)

The objective of this section is to document differences in use case sequence diagrams between previsions made in D7.2 and actual site operation. Differences will be explained in detail, including reasons for the difference, contingency actions taken to match the actual sequence diagram to the planned ones and changes in result evaluation method, when necessary.

2.6 Testing Scenarios

This section provides details about the testing scenarios used for case validation. Testing/validation scenarios should follow specifications set in D7.2. When necessary, deviations from defined scenarios will be documented here.

2.7 Execution timeline

This section provides details of the actual execution timeline of the UC, along the lines of the different trial execution stages defined in the Description of Action (DoA). When necessary, discrepancies from planned execution will be documented, including reason for discrepancy, new execution timeline, and influence of the change in the UC results.

2.8 Intermediate results

In this section, results from trial sites will be summarized, differences from planned results documented, potential effects for project end results described, and introduced corrective actions with planned effects (if any) listed.

3 Trial-site setup and validation process

3.1 Human-Centred Twin Smart Cities Living Lab

The Human-Centred Twin Smart Cities Living Lab includes 3 Use Cases that focus on vehicle and pedestrian traffic in the Jätkäsaari area in Helsinki, Finland.

3.1.1 UC1 – Traffic Flow Prediction & Parking prediction

The Jätkäsaari area is one of the most congested areas in the city of Helsinki and it is a former island, which has been linked to the mainland via a single land connection (a landfill) and one bridge for cars. Jätkäsaari is also the home to the major port facility in Helsinki, where all the passenger car ferries travelling between the cities of Helsinki and Tallinn converge. The port facility serves both passenger and cargo traffic and is considered to be one of the busiest international traffic connections in Europe serving around 4.1 million passengers annually (2020).

The objective of UC1 is to address the traffic-related challenges facing the city of Helsinki and the Jätkäsaari port's managing company, Port of Helsinki (owned by the city) as described in Section 3.1.1.1. To reduce the congestion, efficient prediction of traffic flows and parking availability is planned in these bottleneck locations. The solutions tested in Helsinki are developed to be applicable for other cities addressing their specific traffic situations as they are designed to be scalable and can be replicated in other cities with relative ease. To best address the above development areas, the city of Helsinki has established an ecosystem actor, the Jätkäsaari Mobility Lab[1] to foster innovations and public-private sector collaboration. It offers the Living Lab-environment for the vehicle-focused UC1 and pedestrian-focused UC2.

Several potential business benefits for the stakeholders involved in the UC1 have already been identified, particularly for the sustainability of the use case development and third-party involvement for value-added service development. First, the use of predictive analytics to forecast traffic flows and parking availability in bottleneck areas could lead to significant time and cost savings for individuals and businesses. This benefit is particularly relevant for companies involved in logistics and transportation, as they could optimize their routes and minimize delivery times. Second, the use-case's scalability and replicability in other cities could create a new revenue stream for companies involved in its development and deployment. As other cities seek to address their traffic-related challenges, the Jätkäsaari Mobility Lab's expertise and IoT-NGIN technology could be leveraged to offer value-added services in new markets. Finally, the sustainability of the use-case development and third-party involvement identified in UC1 could result in long-term benefits for companies, as they continue to offer value-added services and maintain their market position.

Together, UC1 and UC2 can provide valuable insights and predictions for traffic flow and parking prediction for a variety of business use cases, helping them make data-driven decisions and improve their operations. Some examples include:

[1] https://mobilitylab.hel.fi

1. Urban Planning and Development: City planners and developers can use the insights from the twin cities living lab to make data-driven decisions about infrastructure development and improvements to traffic flow and parking capacity in different areas of the city.
2. Real Estate: Real estate developers and agents can use the traffic flow and parking prediction data to make informed decisions about property development and investments in different areas of the city based on their accessibility and convenience.
3. Transportation and Logistics: Logistics companies can use traffic flow predictions to optimize their routes and delivery schedules, reducing travel times and improving efficiency. They can also use parking predictions to identify the best parking spots for their delivery vehicles, minimizing the risk of parking violations and fines.
4. Retail and Hospitality: Retailers and restaurants can use the data from the twin cities living lab to determine the best locations for their businesses based on traffic flow and parking accessibility. They can also use the data to identify peak hours and adjust staffing and inventory accordingly.
5. Advertising and Marketing: Marketers can use the data to target consumers with personalized offers based on their location and travel patterns. For example, they can send offers to commuters in congested areas to entice them to stop at a nearby restaurant or store.

In order to reduce congestion through efficient prediction of traffic flows and parking prediction requires the use of advanced technologies and data analysis techniques. Some steps that are taken to achieve this are:

1. Collect Data: The first step is to collect relevant data on traffic flows and parking. This data is collected using various methods such as traffic sensors, GPS data from mobile devices and vehicles, and data from parking meters and cameras.
2. Analyze Data: The next step is to analyze the data to identify patterns and trends in traffic flows and parking demand. This is done by using machine learning algorithms and statistical analysis techniques.
3. Develop Models: Once the data has been analyzed, models are developed to predict traffic flows and parking demand. These models are used to optimize traffic signal timing and parking allocation.
4. Implement Intelligent Transportation Systems (ITS): ITS systems can be used to implement the models developed in step 3. These systems include traffic signal control systems, dynamic message signs, and parking guidance systems.
5. Communicate with Drivers: Drivers are then informed about traffic conditions and parking availability through real-time information systems such as mobile apps and digital signage.
6. Evaluate and Refine: Finally, it is important to evaluate the effectiveness of the measures taken and refine them if necessary. This can be done by monitoring traffic flows and parking demand and adjusting the models and systems as needed.

3.1.1.1 Trial site description

The trial site of the Smart City LL in general is built on top of the Jätkäsaari Smart Junction project that uses sensors and cameras at the bottleneck intersections to collect data on traffic flow, including lane-specific vehicle counts, travel times, and stops, which is then analysed using ML to identify traffic patterns and broadcast to users through the least intrusive means such as TMC systems.

Such sensors, for example, include the radars, located in the areas mapped in the figure below.

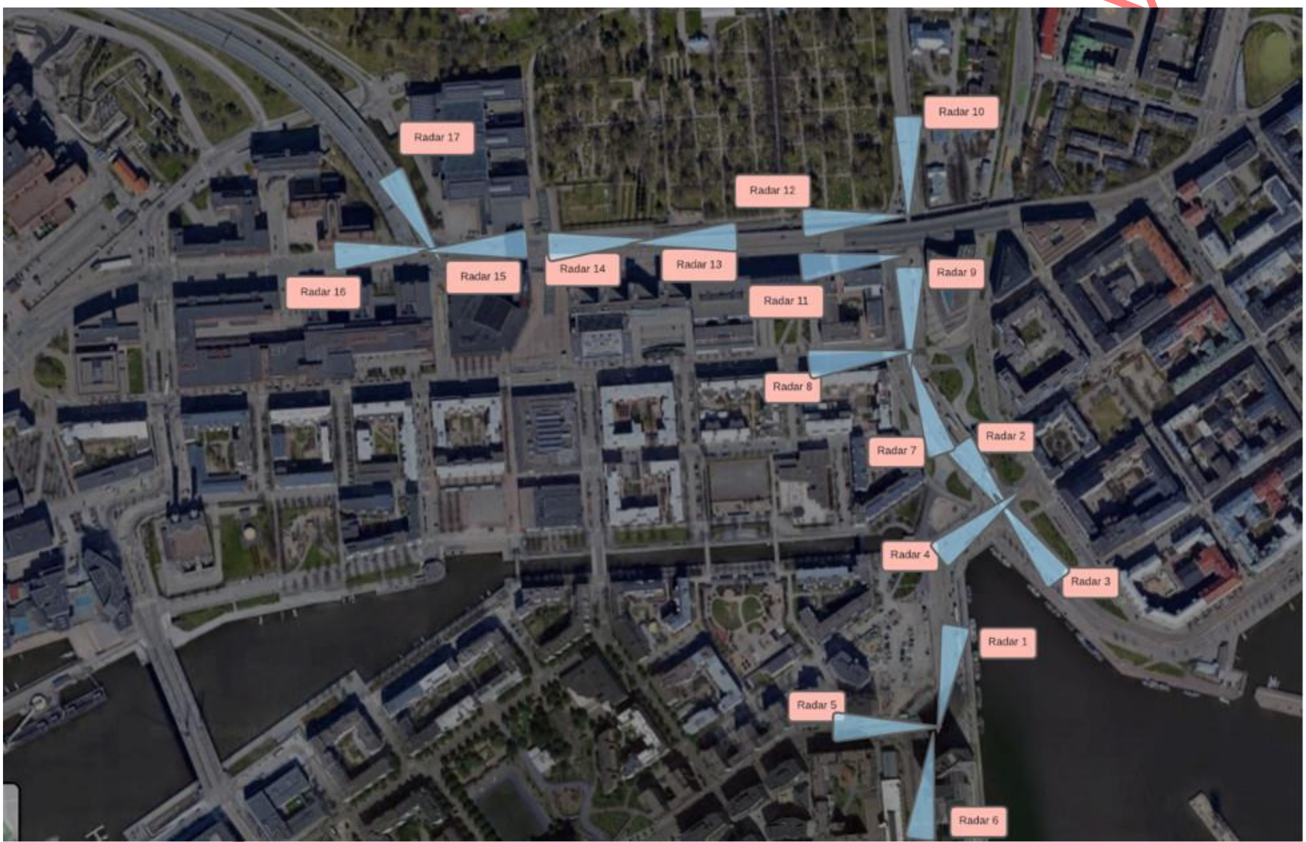


Figure 1: Radar Locations.

Additionally, the infrastructure is equipped with the controllers located in the following areas:

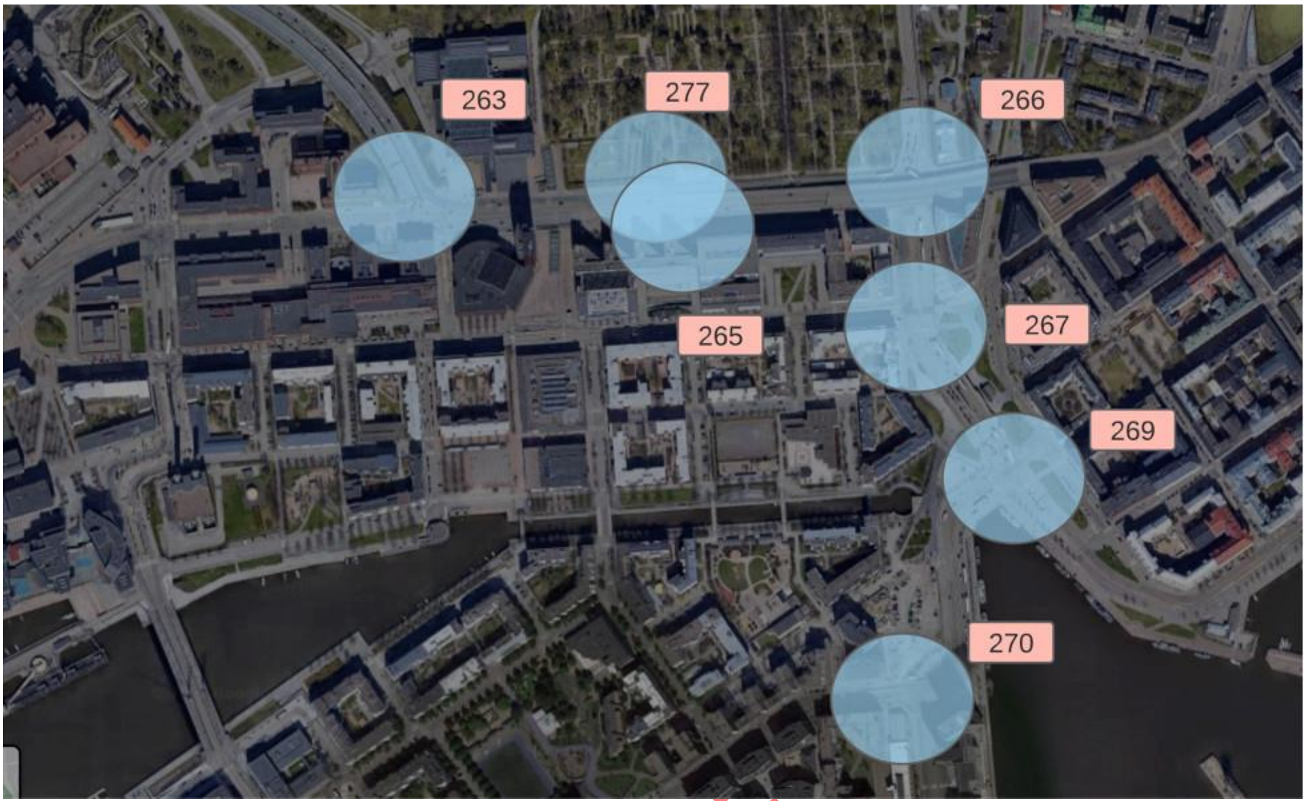


Figure 2: Controller Locations.

Lastly, for the purpose of parking availability estimation, a dataset containing information on parking spaces in the Helsinki region has been utilized. The dataset includes data on the type of parking, validity period, fee information, resident parking zones, parking position, and an estimated number of parking spaces available. The areas covered in the parking dataset can be seen on the map below.

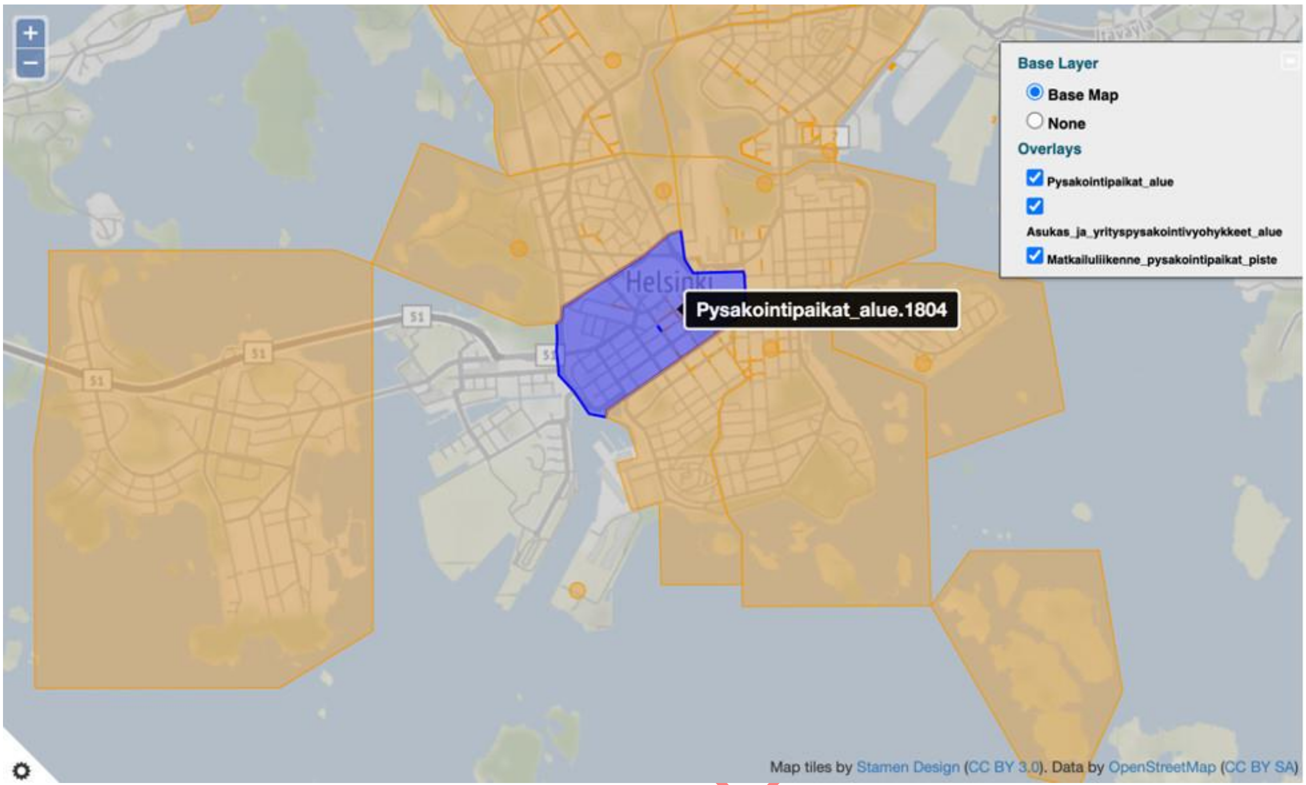


Figure 3: Digitized Helsinki Parking Areas.

3.1.1.2 Required equipment

Table 1 below shows a detailed description of the required equipment for UC1, as already reported in D7.2, along with its type and status (whether it is available or still be procured).

Table 1: Detailed description of the required equipment for UC1.

Equipment	Description / Specifications	Type	Status
Radar	Radar sending data about real-time vehicle positions for traffic management purposes.	sensor	Available
Camera	On-location cameras are installed to monitor the movement of vehicles and pedestrians in the site of the living lab. The produced data will be aggregated to count the number of the different types of objects passing through the living lab location at a given time.	sensor	Available
Signal controller	Traffic signal status	controller	Available
Signal controller	Detector status	sensor	Available

Lidar	Near real-time positions of vehicles, pedestrians and bicycles within the intersection area.	sensor	Available

3.1.1.3 Data collection

At this stage of the use-case the following dataset(s) have been identified as part of the use case:

- **On-location camera feed:** On-location cameras are installed to monitor the movement of vehicles and pedestrians in the site of the living lab. The produced data will be aggregated to count the number of the different types of objects passing through the living lab location at a given time.
- **Radar imagery:** Radar imagery is used to detect cars in a given area and count the totals of different types of vehicles passing through an area at a given time.
- **Parking Places Helsinki:** Digitized parking places data in the city center and resident parking zones, including information on parking type, validity, fees, resident zones, positions, and estimated number of parking spaces.

3.1.1.4 Alignment with IoT-NGIN technologies

Table 2 below shows the IoT-NGIN technologies relevant for UC1. The table has been updated since D7.2 with one new component. For each technology, a description of its role is provided along with any adaptation it might need to be used in the UC, deployment details and related UC requirements and KPIs.

Table 2: Alignment of UC1 with the relevant IoT-NGIN technologies.

WP7 IoT devices configuration demo application	
Description	This demo application integrates SSI technologies to demonstrate how to easily discover, protect, and configure the IoT Triplet (real-world IoT device, Digital Twin, Semantic Twin) while protecting the privacy of the individual users and providing good user experience through low-latency validation. More details can be found in Section 3.1.1.8.
Adaptation and fine-tuning	Configure the app to the IoT Triplets of the use case.
Deployment	The demo application consists of cloud service and mobile application
Related requirements	**REQ_SC1_NF01** – Data integrity **REQ_SC1_NF03** – Privacy of information and security have to be guaranteed
Related KPIs	**KPI_SC_1** – Different types of sensors' data (multispectral/visual camera, RFID) to be analysed > 6 **KPI_SC_5** – Increase Twin Smart Cities IoT infrastructure utilization by at least 30%

3.1.1.5 Use case sequence diagrams

The sequence diagram of the UC1 has not had any modifications since the previous deliverable D7.2.

3.1.1.6 Testing and validation procedures

3.1.1.6.1 Technical validation of the UC

- The key objective of UC1 is to address the traffic issues in Jätkäsaari, which is also reflected in KPI_SC_4, which calls for a traffic congestion reduction of 20%. Reaching that KPI, thus, also validates the use case[2].
- This use case also helps validate the visual recognition functionality of the IoT Device Discovery component.
- The Smart City living lab is used to aid the development of the ST and SSI solutions to ensure they include the necessary properties to aid efficient development, deployment, and interfacing with other systems. This goal is then validated with the other living labs utilising these technologies.

3.1.1.6.2 Validation stages

- The components will first be validated individually in a lab environment during the development process.
- Afterwards, the whole solution will be validated in field conditions.

3.1.1.6.3 User experience testing/validation

- UC1 user experience will be validated with selected questionnaires for the drivers and the port authority. Targeted interviews are conducted to confirm the perceived effect on the business case.
- Helsinki Digital Twin Visualization will be used for the validation of the end-user interface to track congestions, display traffic information, and view best route suggestions. Such validation is intended to be visual.

3.1.1.7 Execution timeline

The updated execution timeline of UC1, divided into multiple phases, is detailed in Table 3 below.

Table 3: Execution timeline of UC1.

Phase	Estimated start date	Estimated end date	Notes
Trial set-up and equipment procurement	Ongoing	June 2022 (M21)	First batch of sensors (cameras and radars) installed and initial calibrations completed.
Initial implementation and validation	M22	M26	Second batch of sensors (e.g., lidars) installed and calibrated. Additional calibrations for the first batch if necessary.

Intermediate implementation and validation	M27	M31	Sensors connected to the simulation model. Installer app was developed and tested.
Final implementation and validation	M32	M36	Linking results between all 3 UCs. Potential business cases to be developed for sustainability.

3.1.1.8 Intermediate results

In the following section, the intermediate results of two development streams of the UC1 Traffic Flow Prediction & Parking Prediction use case will be presented. That includes the development of the Traffic and Parking prediction, and the IoT devices configuration demo application.

3.1.1.8.1 Traffic and Parking prediction

The UC1 Traffic Flow Prediction & Parking Prediction use case has made progress towards one of the primary goals of aiding drivers in choosing less congested routes and times by providing real-time traffic flow information.

To this end, the Smart Junction project has installed various sensors and equipment, such as radars, cameras, and signal controllers, to collect traffic data from the Jätkäsaari area. The collected data includes information about the number of vehicles passing through intersections, their speed, size, direction of movement, and GPS coordinates. This data is used to create real-time traffic flow maps that show the congestion levels in different areas of the city. One of such implementations could be visible in Figure 4.

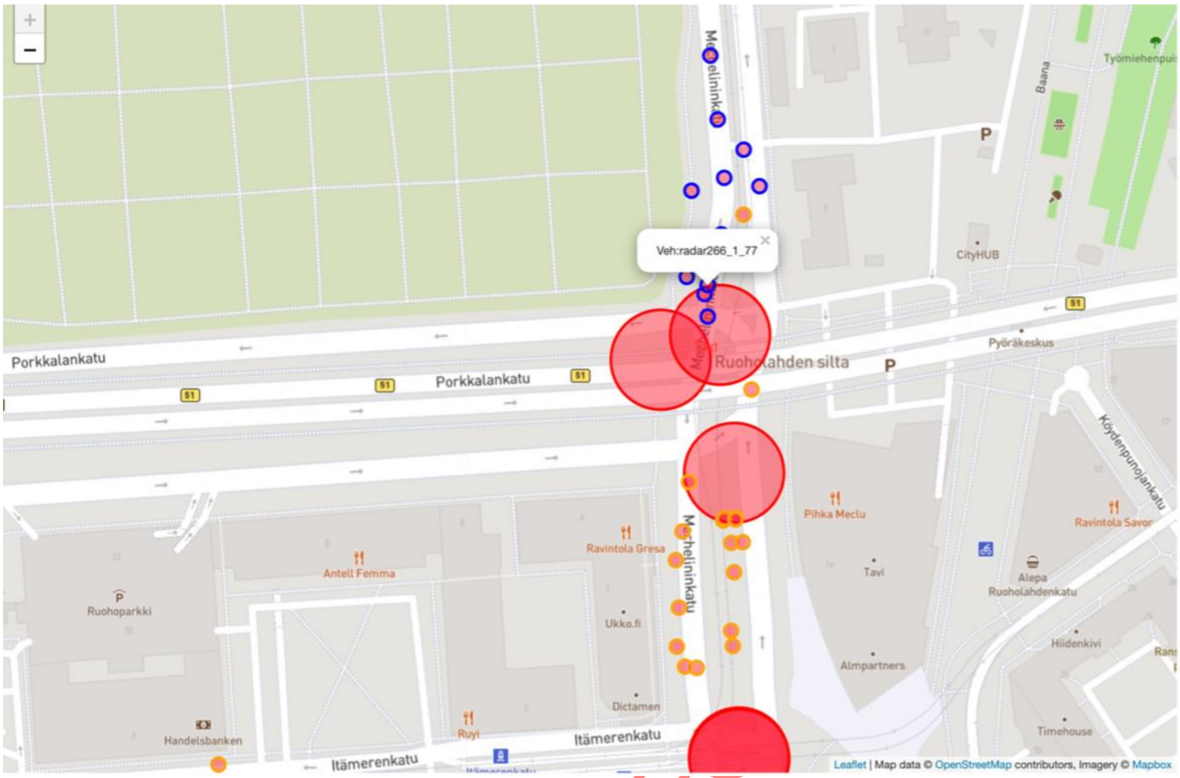


Figure 4: Real-time traffic map segment.

In addition to that, parking data sources available from the city of Helsinki Region Infoshare were identified as a potential complementary data source for the UC1 Traffic Flow Prediction & Parking Prediction use case. This dataset contains information on the location, type, validity period, fee, and estimated number of parking spaces for parking places in Helsinki's city center and resident parking zones. The data has been digitized and drawn by hand based on aerial images, street view images, and traffic control plans of urban planning. While the estimated number of parking spaces is based on the area of the digitized region and may not reflect the actual number of spaces available, the dataset provides valuable information that can be used to enhance the accuracy of parking prediction models. Example of such parking spot availability can be seen at Figure 5.

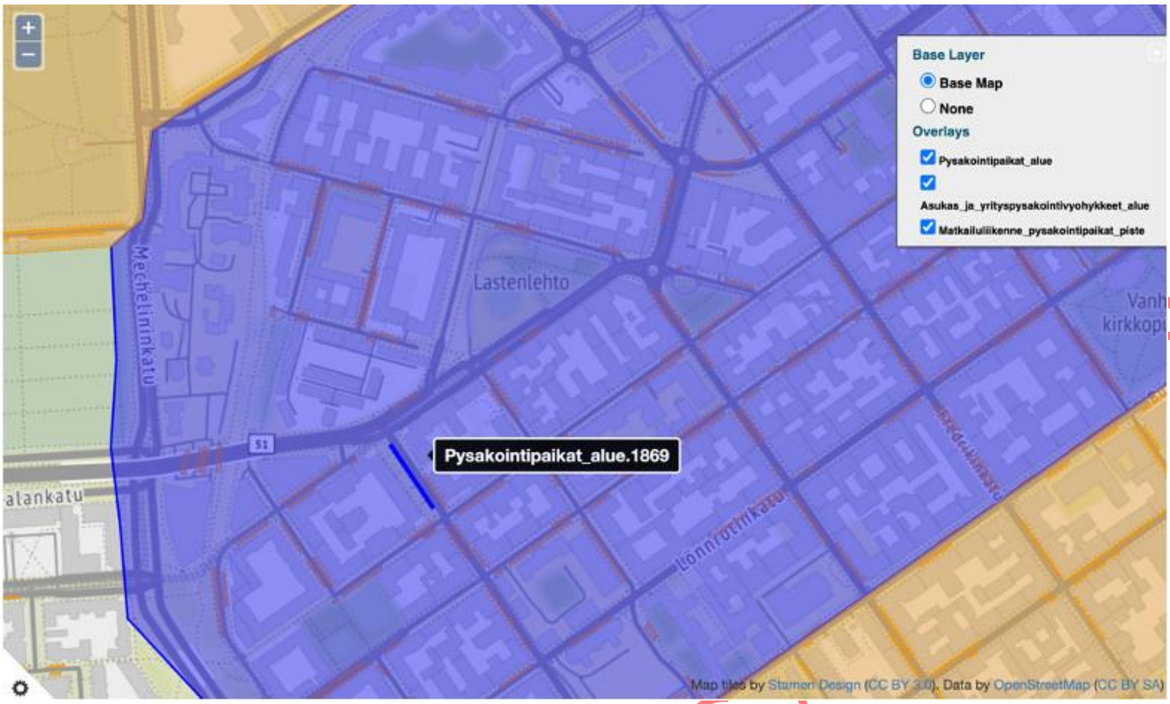


Figure 5: One of the parking segments close to Jätkäsaari.

Several concurrent developments are taking us towards a digital twin of urban mobility. The transportation module of the CityGML 3.0 standard aims to provide a detailed description of the street environment, enabling a comprehensive understanding of the street segments, lanes, pavements, and intersections. The potential of the CityGML 3.0 Transportation model is being explored by Mobility Lab Helsinki. In future applications, information concerning the infrastructure is integrated with real-time traffic counter data and vehicle positions to suggest optimal routes. Using MQTT topic structure, the API for real-time vehicle positions allows end-users to subscribe to relevant messages based on context, such as mode of transport, or route ID. An example of this is shown in Figure 6, showing an interactive visualization combining traffic counter data, real-time vehicle positions, suggested routes, and the 3D description of the Urban environment. Such interfaces can combine suggested routes with incident and roadwork points.

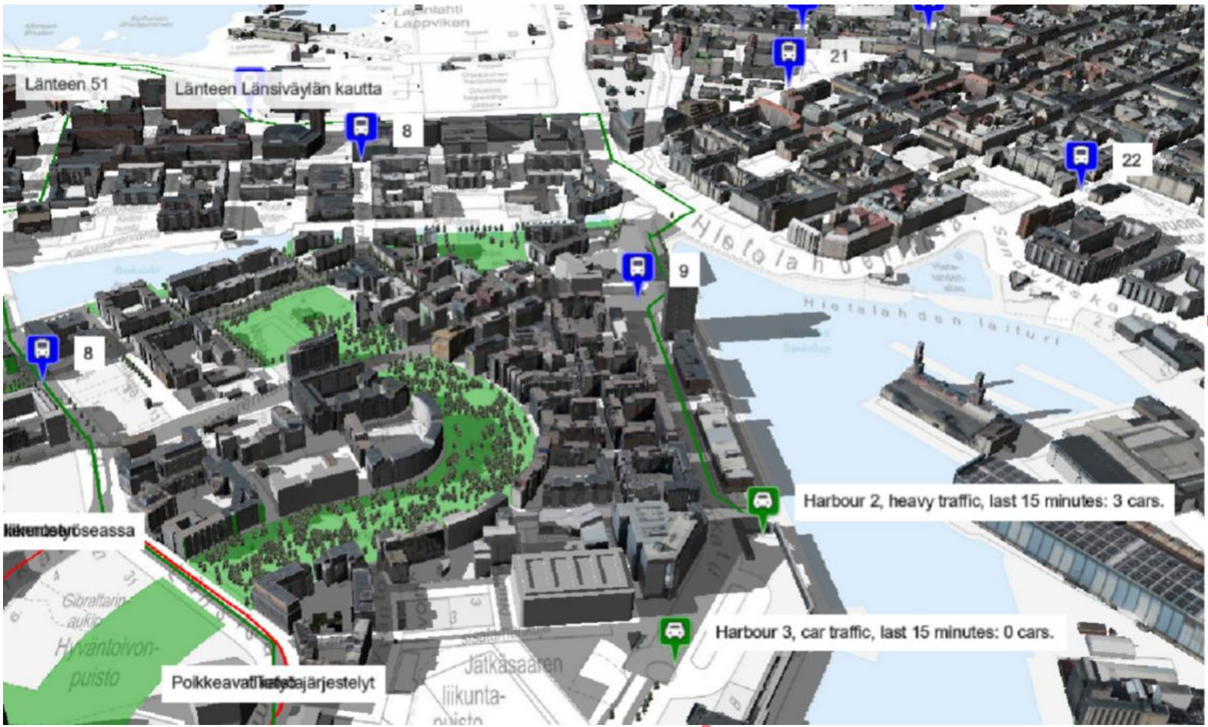


Figure 6: Demo of an interactive visualization combining traffic counter data, real-time vehicle positions, suggested routes and the 3D description of the Urban environment (showcasing part covered by Smart Junction).

As the project's UC1 approaches its final stages, several next steps have been identified to further enhance the traffic flow and parking prediction use case:

- Establish a connection between the existing traffic data and identified parking data to enable better parking predictability by leveraging the insights from both data sources.
- Develop an ML model to predict congestion levels based on vehicle movement using IoT-NGIN components.
- Finalize the data integration process to the Urban Open Platform.
- Identify methods for integrating the results of the data analysis into third-party driver applications.

These next steps will be the primary focus of the upcoming deliverable, D7.4, which will report on the progress and achievements of the UC1.

3.1.1.8.2 IoT devices configuration demo application

To support the deployment of sensors in the Jätkäsaari Living lab, a configuration demo application has been developed to demonstrates how to easily discover, protect, and configure the IoT Triplet while protecting the privacy of the individual users and providing good user experience through low-latency validation. The key actors of the demo use case are illustrated in Figure 7.

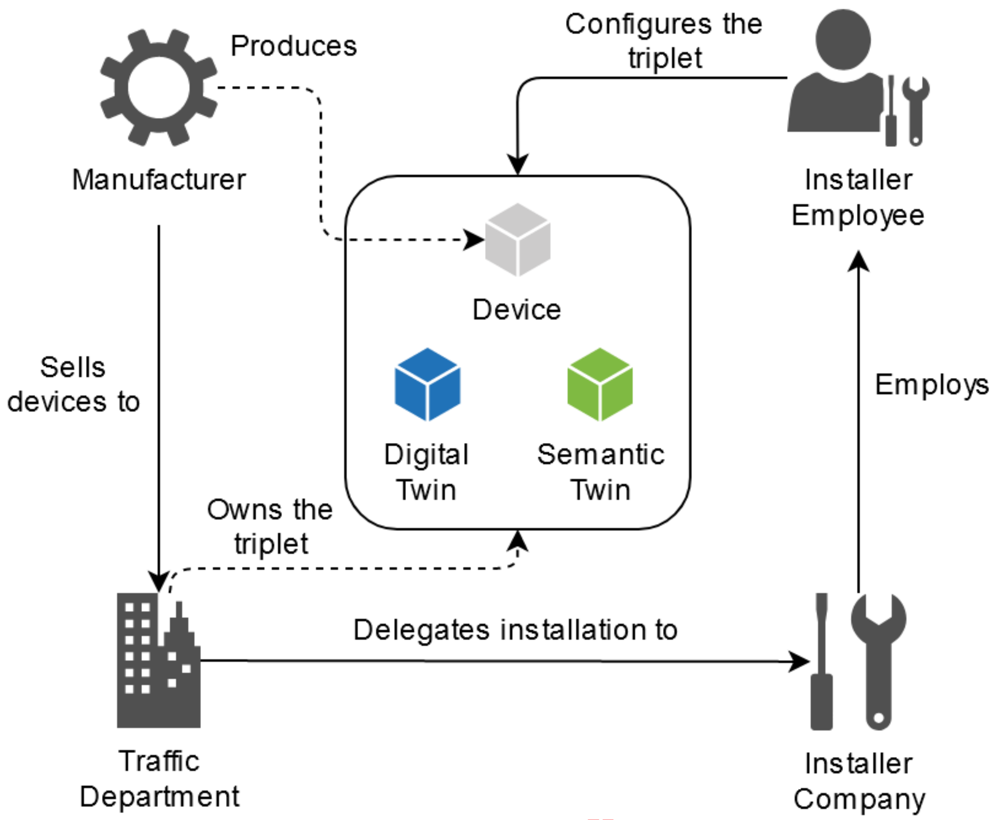


Figure 7: Summary of the IoT devices configuration demo application.

The Traffic Department of a City buys IoT Devices from a Manufacturer and wants to install them to a Smart City project. The Traffic Department initialises a device's Digital Twin and Semantic Twin with the basic information required to delegate the setup to an external Installer Company. Moreover, the Traffic Department creates a QR code for each device, embedding a GTIN number that, once resolved by a GS1 Digital Link Resolver server, provides the locations to access the device's Digital Twin and Semantic Twin.

The Installer Company employs one or more Installer Employees to go around the city and install the devices (and possibly maintain them afterwards). To finalise the installation of a device, an Installer Employee accesses the Digital Twin and the Semantic Twin of that device. To access them, they require a credential that can be obtained from the Installer Company Authorization Server. The Employee scans the QR code on the device with a mobile phone application. Once scanned, the QR code redirects the Employee to the Semantic Twin. At access request, the server hosting the Semantic Twin begins an access control protocol to know the privileges of the person who is requesting the access. With the QR code being accessible to anyone, any citizen of the City could potentially get access to the Semantic Twin to view information about the Device.

With this demo, we address the following problems:

- Discovering the Twins related to an IoT Device.
- Enabling secure access to the triplet.
- Trusting the data received by the triplet.
- Protecting people's privacy.

- Detecting malicious activities on the triplet.

Additionally, as an intermediate result, business cases were developed demonstrating the benefits of UC1 in various scenarios. These cases, available in Section 3.1.1, showcase the potential impact of using data-driven insights to optimize traffic flows, parking allocation, and co-commuting solutions. These business cases provide tangible examples of how UC1 components can bring value to different contexts and can help stakeholders explore the potential of data-driven insights in transportation.

3.1.2UC2 – Crowd Management

As described in UC1, the Jätkäsaari area in the city of Helsinki poses several key challenges for the city of Helsinki due to its geographical location and the high density of traffic flows in the area. UC2 demonstrates the use of open data, user data, and IoT data on crowd management. Of these, the role of IoT data for crowd management purposes is the most advanced and is now fully operational in the Jätkäsaari area. There, the Port of Helsinki company and HSL (Helsinki Regional Transport) are potential Helsinki-based business case partners.

In the IoT-driven UC2, traffic fluency will be monitored and processed via cameras and radars installed at bottleneck intersections to achieve crowd steering based on the application of AI. UC2 is also considering crowd management monitoring of the busy Helsinki-Tallinn cross-border commuter harbour during daily peak hours and any unexpected events based on predictive algorithms.

The role of open data and user data in UC2 closely follows the development progress of the *Urban Open Platform* conducted by the cross-border FinEst Twins-project[3] between the cities of Helsinki and Tallinn, operated by the FinEst Centre based in Estonia. The FinEst Centre focuses on mobility, energy, and the built environment working together by governance and urban data analytics. The centre develops cross-border knowledge transfer infrastructure and acts as a springboard for the exportation of Finnish-Estonian knowledge and combined service solutions on a global scale.

Various business benefits were determined during the UC2 development. Firstly, by using anonymous monitoring to identify and manage bottlenecks, the UC2 trial has the potential to improve the overall customer experience for those travelling to and from the Jätkäsaari harbour. This can lead to increased customer satisfaction and loyalty. Secondly, The UC2 trial aims to improve crowd and traffic management, as well as parking space management, which leads to increased efficiency in the transport system. This can result in cost savings for the Port of Helsinki and Helsinki Regional Transport Authority, as well as increased revenue through better utilization of existing resources. Lastly, as with UC1, the solutions developed in the UC2 trial are designed to be scalable and replicable in other cities facing similar traffic-related challenges. This can result in a larger impact on improving traffic flow and the customer experience in urban areas beyond Helsinki.

UC2 can be used for crowd management in various business use cases, including:

1. Large Events: The living lab can be used to manage large events such as concerts, festivals, and sporting events. By using sensors and real-time data, organizers can monitor crowd flow and adjust the flow of people to prevent overcrowding and ensure the safety of participants.

2. Public Transportation: The living lab can also be used to manage crowds in public transportation systems such as buses and trains. The data from the sensors can be used to optimize routes, adjust schedules, and monitor capacity to ensure that there are no overcrowding or delays.
3. Retail Stores: In retail stores, the living lab can be used to manage crowds during sales events, holidays, or special promotions. The data collected can be used to optimize store layouts, adjust staffing levels, and provide a better shopping experience for customers.
4. Smart Cities: Human-centered twin smart cities living lab can be used to manage crowds in smart cities. The data collected can be used to optimize traffic flow, adjust public transportation routes, and monitor air quality to ensure the health and safety of citizens.
5. Emergency Services: The living lab can also be used to manage crowds during emergency situations such as natural disasters, terrorist attacks, or public health emergencies. By using real-time data, emergency responders can quickly and efficiently deploy resources to areas where they are needed most.

Overall, the Human-Centered Twin Smart Cities Living Lab provides an innovative approach to crowd management that can help businesses improve their operations, enhance customer experiences, and ensure the safety of their customers and employees.

IoT (Internet of Things) devices can provide a wealth of data that can be used to improve crowd management. Here are some ways to use IoT data for better crowd management:

1. Real-time monitoring: IoT sensors can be used to monitor crowds in real-time, providing insights into the number of people, their movements, and their behavior. This data can be used to identify potential congestion points and adjust traffic flow accordingly.
2. Predictive analytics: By analyzing historical data from IoT sensors, you can predict crowd behavior and adjust crowd management strategies accordingly. This can help you identify potential problems before they occur and take proactive measures to prevent them.
3. Location-based services: By using IoT devices to track the location of people in a crowd, you can provide targeted services and information to individuals based on their location. For example, you can direct people to less crowded areas, provide directions to emergency exits, or offer promotions to people in certain areas.
4. Automated alerts: IoT sensors can be used to detect potential safety hazards, such as overcrowding or fire hazards. Automated alerts can be sent to crowd management personnel, allowing them to quickly respond and take appropriate action.

Overall, IoT data can be a valuable tool for improving crowd management, providing real-time insights and predictive analytics that can help you manage crowds more effectively and efficiently.

3.1.2.1 Trial site description

The trial site at Jätkäsaari harbour in Helsinki, Finland was described in D7.2 document, outlining the objectives of the efficient crowd and traffic management, parking space management, and improved access to public transport. The UC2 trial aims to provide new insights into crowd movement fluency and to develop solutions to manage bottlenecks and improve the overall transportation experience in the Jätkäsaari harbour environment.

3.1.2.2 Required equipment

Table 4 below shows a detailed description of the required equipment for UC2, as already reported in D7.2, along with its type and status (whether it is available or still be procured).

Table 4: Detailed description of the required equipment for UC2.

Equipment	Description / Specifications	Type	Status
Radar	Radar sending data about real-time vehicle positions for traffic management purposes.	sensor	Available
Camera	On-location cameras are installed to monitor the movement of vehicles and pedestrians in the site of the living lab. The produced data will be aggregated to count the number of the different types of objects passing through the living lab location at a given time.	sensor	Available
Signal controller	Traffic signal status	controller	Available
Signal controller	Detector status	sensor	Available
Lidar	Near real-time positions of vehicles, pedestrians and bicycles within the intersection area.	sensor	Available

3.1.2.3 Data collection

The following updated list of dataset(s) have been identified as part of the use case:

- **On-location camera feed:** On-location cameras are installed to monitor the movement of vehicles and pedestrians in the site of the living lab. The produced data will be aggregated to count the number of the different types of objects passing through the living lab location at a given time.
- **Radar imagery:** Radar imagery is used to detect cars in a given area and count the totals of different types of vehicles passing through an area at a given time.
- **UC3 data:** Twitter data collected and analyzed to predict traffic demand
- **Marine traffic data**: Data that includes vessel location information, harbor schedules, and marine warnings.

3.1.2.4 Alignment with IoT-NGIN technologies

Table 5 below shows the IoT-NGIN technologies relevant for UC2. The table has been updated since D7.2 with one new component. For each technology, a description of its role is provided along with any adaptation it might need to be used in the UC, deployment details and related UC requirements and KPIs.

Table 5: Alignment of UC2 with the relevant IoT-NGIN technologies.

WP7 – IoT devices configuration demo application	
Description	This demo application integrates SSI technologies to demonstrate how to easily discover, protect, and configure the IoT Triplet (real-world IoT device, Digital Twin, Semantic Twin) while protecting the privacy of the individual users and providing good user experience through low-latency validation. See Section 3.1.1.8 for more details.
Adaptation and fine-tuning	N/A
Deployment	The demo application consists of cloud service and mobile application
Related requirements	**REQ_SC1_NF01** – Data integrity **REQ_SC1_NF03** – Privacy of information and security have to be guaranteed
Related KPIs	**KPI_SC_1** – Different types of sensors' data (multispectral/visual camera, RFID) to be analysed > 6 **KPI_SC_5** – Increase Twin Smart Cities IoT infrastructure utilization by at least 30%

3.1.2.5 Use case sequence diagrams

The sequence diagram of the UC2 has not had any modifications since the previous deliverable D7.2.

3.1.2.6 Testing and validation procedures

3.1.2.6.1 Technical validation of the UC

The key objective of UC2 is to address the crowd management issues in Jätkäsaari and the harbour terminal. This objective is related to the KPI_SC_4, which considers traffic congestion reduction, and the aim is to evaluate the reduction of crowds as part of evaluating this KPI.

This use case also helps validate the visual recognition functionality of the IoT Device Discovery component.

The Smart City living lab is used to aid the development of the ST and SSI solutions to ensure they include the necessary properties to aid efficient development, deployment, and interfacing with other systems. This goal is then validated with the other living labs utilizing these technologies.

3.1.2.6.2 Validation stages

There are two stages:

1. The components will be first validated individually in a lab environment during the development process.
2. Then the whole solution will be validated in field conditions.

3.1.2.6.3 User experience testing/validation

UC2 user experience will be validated with selected questionnaires for the pedestrians and the port authority. Targeted interviews will be conducted to confirm the perceived effect on the business case.

Test scenario

The testing steps include:

1. Collect beacon or heatmap data to monitor crowd movement.
2. Collect Marine traffic data to estimate the arrival time of the upcoming ferries
3. Utilize UC3 data to additionally estimate the demand for transport
4. Harmonize the data for analysis
5. Create an ML model to compare demand with supply of the transport system.
6. Presenting result in a form suitable for the possible end-user (transport availability, approximate wait times, better route, etc.)
7. Test the accuracy of the prediction by comparing the predicted data with actual figures.
8. Test the integration of the result data to 3rd party endpoint.

3.1.2.7 Execution timeline

The execution timeline of UC2, as already reported in D7.2, divided into multiple phases, is detailed in Table 6.

Table 6: Execution timeline of UC2.

Phase	Estimated start date	Estimated end date	Notes
Trial set-up and equipment procurement	Ongoing	June 2022 (M21)	First batch of sensors (cameras and radars) installed and initial calibrations completed.
Initial implementation and validation	M22	M26	Second batch of sensors (e.g., lidars) installed and calibrated. Additional calibrations for first batch if necessary.
Intermediate implementation and validation	M27	M31	Sensors connected to simulation model.
Final implementation and validation	M32	M36	Linking results between all 3 UCs. Potential business cases to be developed for sustainability.

3.1.2.8 Intermediate results

This section presents the intermediate results of UC2, which focus on crowd management optimization. The following results have been obtained so far:

- Cameras and other sensors, such as Bluetooth beacons, have been deployed to gather data on crowd movements in public spaces.
- ML modeling is being planned in synergy with UC3 to detect crowds and predict potential bottlenecks. By leveraging the Twitter data analyzed in UC3, UC2 can improve its accuracy in predicting crowd movements and potential congestion.
- Potential solutions for crowd control are explored, including the use of alternative routes and notifications to guide crowds through third-party applications.
- To ensure efficient crowd management, it is crucial to take into account the arrival of passengers from ferries, which heavily impacts crowd movement, especially in the harbor region of Jätkasääri.

During the project, it was discovered that Bluetooth beacons were a valuable source of data for identifying the specific locations of human crowdedness, the result could be seen on the Figure below. The specific location is the entrance to the harbour. While the beacons at the moment could not provide information about the movement of crowds, they were able to provide insights into the specific areas where large numbers of people were gathered.

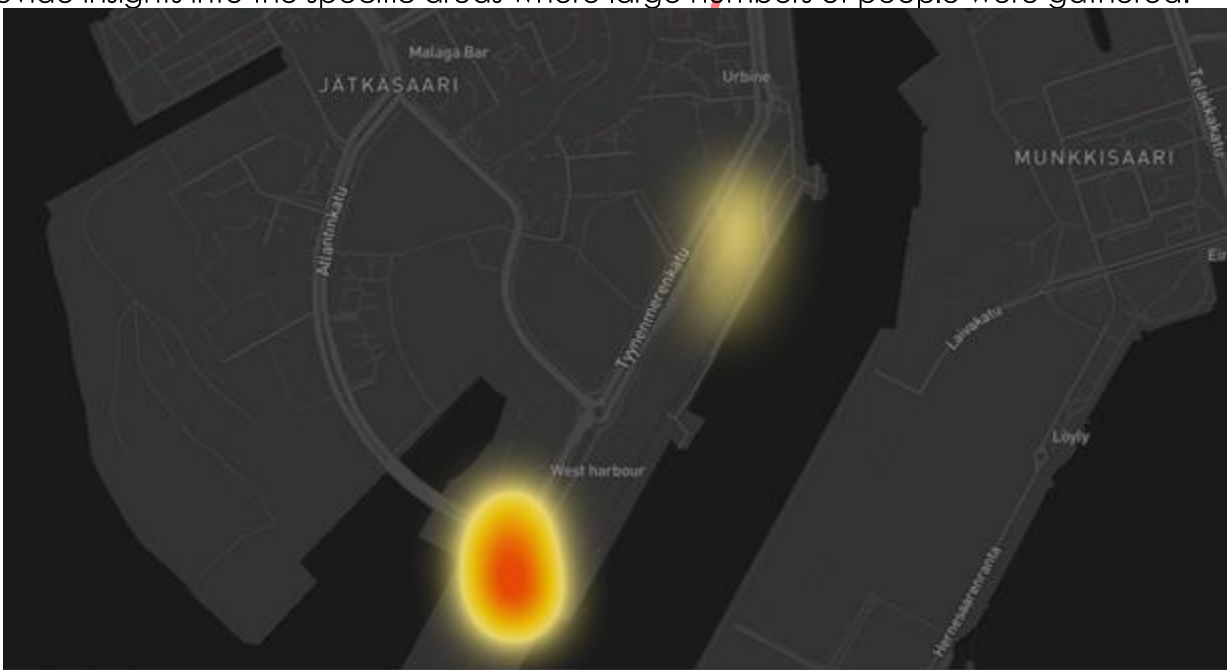


Figure 8: Heatmap based on Bluetooth data from Jätkäsaari harbour area, which highlights the need for efficient crowd management in the harbour terminal (the red hotspot). Screenshot from website made by Hypercell (https://www.heatmap.fi/helsinki/).

To enhance the accuracy of crowd prediction, UC3 data is incorporated into UC2. Twitter posts are manually generated by the volunteers. Initial tests of the Twitter APIv2 were made to explore the features and topics of every post. More detailed overview of the data collection is described in 3.1.3.3. The integration of UC3 data into UC2 logic may provide valuable measures of crowd numbers. For instance, by analysing Twitter data, the ML model is planned to identify events or situations that may attract crowds, like concerts or public rallies. This information can be used to better predict crowd movements and potential bottlenecks. Twitter data can also provide insights into traffic congestion and transportation delays, which can have an impact on crowd movements in public spaces.

Additionally, as an intermediate result, business cases were developed demonstrating the benefits of UC2 in various scenarios. These cases, available in Section 3.1.2, showcase the potential impact of using data-driven insights to optimize traffic flows, parking allocation, and co-commuting solutions. These business cases provide tangible examples of how UC2 components can bring value to different contexts and can help stakeholders explore the potential of data-driven insights in transportation.

In addition, as Jätkäsaari is a harbor region, the movement of crowds heavily depends on the arrival of passengers from ferries. To optimize crowd management in this context, potential solutions were explored, including the integration of marine traffic data provided by Helsinki Digitraffic. This data includes marine warnings, harbor schedules (gathered from the Portnet-system), and vessel location AIS (Automatic Identification System) information. By incorporating some of the marine traffic data, the UC aims to improve the accuracy of predicting when crowds will arrive, and therefore enhance the effectiveness of crowd control solutions such as alternative route notifications to the 3rd party apps.

Lastly, UC2 also uses IoT devices configuration demo application, which was described in Section 3.1.2.

3.1.3UC3 – Co-commuting solutions based on social networks

UC3 aims to showcase the application of IoT-NGIN technologies in providing co-commuting solutions at the neighbourhood and cross-border levels by combining IoT data with virtual citizen-generated IoT data from social networks. The solution will leverage the Urban Open Platform and Lab (UOP.Lab) developed in the Finest Twins project, which provides a framework for knowledge and tools transfer to SMEs. The use case will extend the functionality of the UOP.Lab by incorporating IoT data from social networks to enable co-commuting solutions based on social networks.

The UC3 activity on co-commuting solutions is linked to the local crowd management activity of UC2, where local business partners are also seeking solutions for a combined, co-commuting offer to enhance efficient passenger traffic flows, and to UC1 to further help reduce traffic congestion in the Jätkäsaari area.

UC3 in close co-operation with the developing Urban Open Platform (UOP) of FinEst Twins Project (H2020 856602) seek ways for trusted and intelligent traffic information management and sharing, which facilitates safe co-commuting solutions. The traffic information is shared via the information systems on-board the passenger ferries and at the passenger terminals in Helsinki and Tallinn, and the co-commuters can then use their mobile apps to request or offer rides.

The UC3 co-commuting solutions offers several potential business benefits. Firstly, the use of social media analytics to predict demand for co-commuting services can help improve the efficiency of passenger traffic flows and reduce traffic congestion in the city. The use of social network data feeds can additionally help to replace hardware-driven methods to detect crowdedness with software-driven methods, reducing the cost of such detection, and thus generating more traffic data with minimum resources. Secondly, the IoT-NGIN components together with the Urban Open Platform (UoP) provide third-party organizations access to co-commuting data, providing valuable insights into demand patterns, allowing 3rd party applications to better serve their customers and optimize their operations, and as a result creating new revenue streams for the companies involved. Finally, the integration of various third-party applications, including ridesharing apps, ferry company apps, and scooter apps enables the creation of a more comprehensive picture of transportation demand and patterns. This allows businesses to better understand the needs of their customers and make informed decisions about service offerings, routes, and resources. By sharing data and insights, these stakeholders can work together to provide more seamless and efficient transportation options for customers.

UC3 can have several business use cases, some of which are listed below:

1. Ride-sharing services: One of the primary business use cases of this living lab is to facilitate ride-sharing services. By analyzing social network data and co-commuting patterns, companies can identify potential ride-sharing partners for commuters. This can help reduce traffic congestion and lower transportation costs for both individuals and companies.
2. Advertising and Marketing: Companies can use the living lab to gain insights into commuter behavior and preferences. This data can be used to target advertising and marketing campaigns more effectively. For example, if the data indicates that a significant number of commuters are interested in eco-friendly transportation options, companies can create marketing campaigns highlighting their eco-friendly credentials.
3. Mobility as a Service (MaaS): MaaS is a transportation model that combines various transportation options, such as public transit, ridesharing, and bike-sharing, into a single platform. By analyzing social network data and co-commuting patterns, companies can identify the most effective MaaS options for commuters. This can help reduce transportation costs and improve overall transportation efficiency.
4. Workplace location optimization: Using social network data and co-commuting patterns, companies can identify the optimal location for their workplaces. By locating their workplaces in areas that are convenient for their employees to commute to, companies can improve employee satisfaction and reduce turnover.
5. Transportation infrastructure planning: The living lab can be used to identify transportation infrastructure needs based on co-commuting patterns. For example, if the data indicates that a significant number of commuters use a particular route or mode of transportation, companies and governments can invest in infrastructure improvements in that area.
6. Corporate Commuting: The living lab can be used to develop co-commuting solutions for employees of large corporations. By leveraging social networks and real-time data, the living lab can help employees find ride-sharing opportunities with their colleagues, which can reduce traffic congestion, lower transportation costs, and improve employee satisfaction.
7. Public Transportation: The living lab can also be used to develop co-commuting solutions for public transportation systems such as buses and trains. By analyzing data from social networks, the living lab can help passengers find co-commuting

opportunities that are more convenient and efficient, which can improve ridership and reduce traffic congestion.

8. Carpooling Services: The living lab can be used to develop co-commuting solutions for carpooling services such as UberPOOL and Lyft Line. By leveraging social networks and real-time data, carpooling services can better match riders with compatible co-commuters, which can increase the efficiency of the service and reduce wait times for riders.

Overall, the Human-Centered Twin Smart Cities Living Lab for Co-commuting solutions based on social networks can be a powerful tool for companies and governments looking to improve transportation efficiency and reduce transportation costs.

Co-commuting solutions, which are linked to local crowd management activities and that collaborate with local business partners can create a combined, co-commuting offer that enhances efficient passenger traffic flows and reduces traffic congestion. Some steps to achieve this are listed below:

1. Identify local business partners: Work with local businesses such as restaurants, shops, and entertainment venues to identify areas of collaboration. They may be interested in promoting co-commuting solutions to their customers, such as offering discounts for those who use public transportation or carpooling.
2. Analyze crowd management data: Use IoT sensors to collect data on crowd movements, traffic flows, and congestion points. Analyze this data to identify areas that could benefit from co-commuting solutions, such as bus or shuttle services that connect parking lots to major transportation hubs.
3. Develop a co-commuting offer: Based on the analysis of crowd management data, work with local transportation providers to develop a co-commuting offer that enhances passenger traffic flows. This could include shuttle services from parking lots to transportation hubs or discounts for those who use public transportation or carpooling.
4. Promote the co-commuting offer: Collaborate with local business partners to promote the co-commuting offer to their customers. This can be done through social media, email newsletters, and other marketing channels.
5. Monitor and adjust: Monitor the effectiveness of the co-commuting offer and adjust as needed based on feedback and data analysis. This can help ensure that the offer continues to meet the needs of the community and contributes to reduced traffic congestion.

By leveraging co-commuting solutions and collaborating with local business partners, it is possible to create a combined offer that enhances efficient passenger traffic flows and reduces traffic congestion.

3.1.3.1 Trial site description or update

The UC3 is built on top of UC1 and UC2 and mainly focuses on the Jätkäsaari district. UC3 will further enrich the awareness of transportation needs by utilising social media, especially

Twitter. The UoP has the capability to provide stream processing functionalities that can contain advanced analytics such as Natural Language Processing (NLP). The advanced analytics together with the information from UC2 will be enabling the future development of co-commuting services that benefit from predictive demand recognition.

The scope of the UC3 is to demonstrate the data flow and provide the proof of concept of such service as feasible. The scope of the project does not include a ride-sharing service so the UC3 will focus on the information flows used to demonstrate the potential benefit of the IoT-NGIN concept on advanced stream processing in the mobility domain.

Figure 9 depicted partly from the D6.2 gives a more detailed view on the scope of UC3.

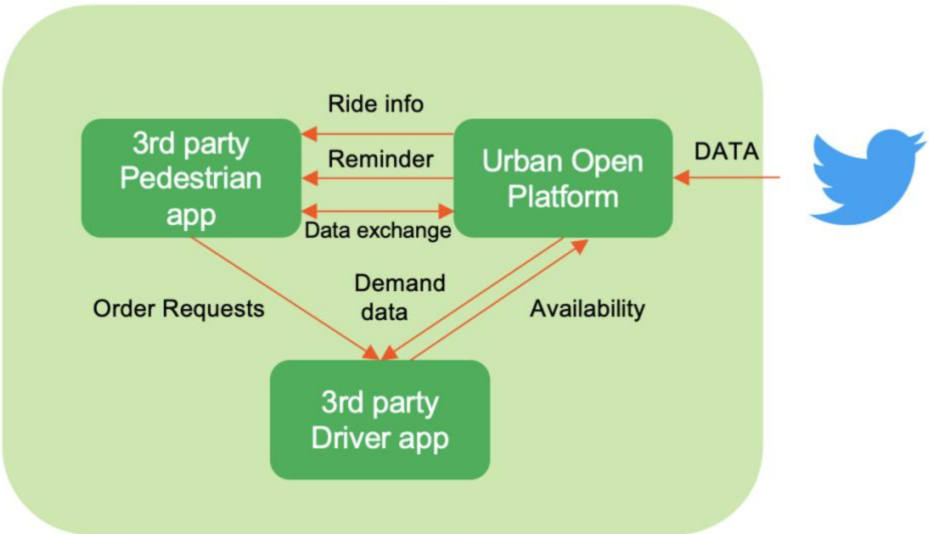


Figure 9: Part of UC3 architecture instantiation.

The strategy for this use case involves collecting data from the Twitter feed and feeding it into Urban Open Platform (UoP). The tweets are then analysed to create a demand prediction model, and the resulting information are then be shared with various third-party applications, including ridesharing apps, ferry company apps, and scooter apps, among others.

In the scope of UC3 Urban Open Platform, provides third-party organizations access to co-commuting data. UoP collects sensor data from cross-border cities, city systems, and public and private sector city infrastructure. In the context of UC3, the UoP acts as the data processing layer for the Co-commuting. It integrates data, exchanges training data sets, communicates with third-party applications, and can generate high-demand reminders between sub-systems to ensure seamless operation. Figure 10 provides a more detailed representation of the technical components that underpin the Urban Open Platform (UoP).

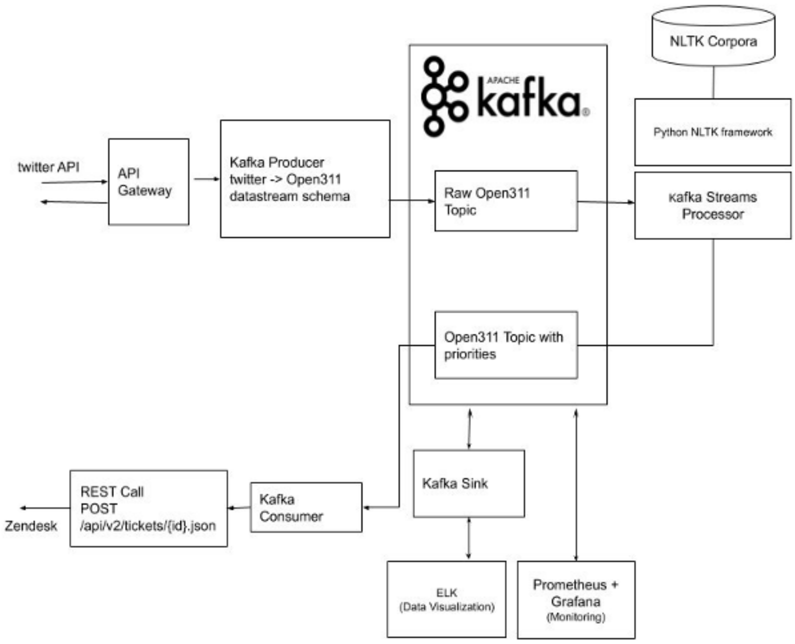


Figure 10: Main technical components involved in UC3 under UoP.

3.1.3.2 Used equipment

Table 7 below shows an updated description of the required equipment for UC3, along with its type and status (whether it is available or still be procured).

Table 7: Detailed description of the required equipment for UC3.

Equipment	Description / Specifications	Type	Status
Commuters' mobile device	The mobile devices of commuters in the use case trial site area, from which data will be collected.	Mobile device	To be procured
Bluetooth beacons	Used to measure actual crowdedness at the harbours.	Device	They are available as part of UC2 but also as part of Helsinki Region Infoshare
Information displays	The information displays at the Harbour Terminals and on ferries from which data will be collected.	Displays	Available from the harbours
Edge server	Capable of running IOT-NGIN components		Available as part of UoP

3.1.3.3 Data collection

In UC3 the social network data feeds of the volunteers in the living lab will be collected to retrieve routes, commute times and features of the provided commuting solution. Twitter was chosen to be the social network under analysis. The following data covered in Table 8 is planned to be retrieved using Twitter API v2.

Table 8: Twitter data collected in UC3.

Data	Description
Twitter objects /Hashtag messages	Each Tweet consists of a variety of entities. Such entities are parsed out of the Tweet text as a JSON object. For UC3 hashtag objects will be requested.
Tweet lookup endpoint	Provides Tweet metadata returning most recent post including such fields as tweet text, timestamp, metadata of hashtag, etc.
Tweet object expansion /place.fields	Twitter posts containing following field values and types: • Geo – Contains place details in GeoJSON format • Place type – Specified particular type of information represented in place information, such as city name or point of interest • Full name – A longer-form detailed place name
Tweet annotations - entity / context	Twitter annotations provide contextual information about the tweet. There are two known annotation types: • Entity annotation – delivering entity payload including people, places, products, organizations • Context annotation – in pair with entity annotation provides tweet topics allowing tweet categorization. There are more than 80 topics available. At least three are applicable to UC3: o /countries, /cities o /travel

The Twitter API v2 offers a variety of endpoints that provide access to different data types, such as tweets, users, search results, trends, and more. Furthermore, within each endpoint, there are often multiple parameters that can be used to filter or customize the data that is returned. For example, the tweet endpoint includes parameters for filtering by keyword, location, date range, or language.

Therefore, it is considered that each endpoint and its associated parameters are potential data sources that can be analysed as part of **KPI_SC_2.**

Twitter feed data can be complemented with a better measurable crowdedness data coming from the Helsinki Region Infoshare, detailed in Table 9.

Table 9: Sensor data collected within UC3.

Data	Description
Beacon data	Bluetooth beacons are installed on the stops in HSL area. Such data represents the crowdedness at spot and can therefore be used to check the quality of the crowdedness data coming from the twitter feed analysis.

3.1.3.4 Alignment with IoT-NGIN technologies update

There are no updates from the previous deliverable D7.2 about the alignment with IoT-NGIN technologies for the UC3.

3.1.3.5 Use case sequence diagram(s)

The main steps in the sequence diagram of this use case can be summarized as follows:

- UoP is gathering data from the volunteer's Twitter feed and possibly from the HSL beacons.
- Data gathered will then be anonymized.
- ML detects large events, crowdedness, or other possible indications of the crowd movements. Additionally, data from the HRI beacons is harmonized with the Twitter feed based crowdedness data.
- In case of increased demand identification, reminder of the possible ride sharing option is sent to the 3rd party pedestrian app.
- In case of increased demand identification, reminder of the possible ride sharing demand is sent to the 3rd party driver app.

Figure 11 below shows a sequence diagram of UC3, depicting the involved actors and their interactions with the relevant IoT-NGIN technologies.

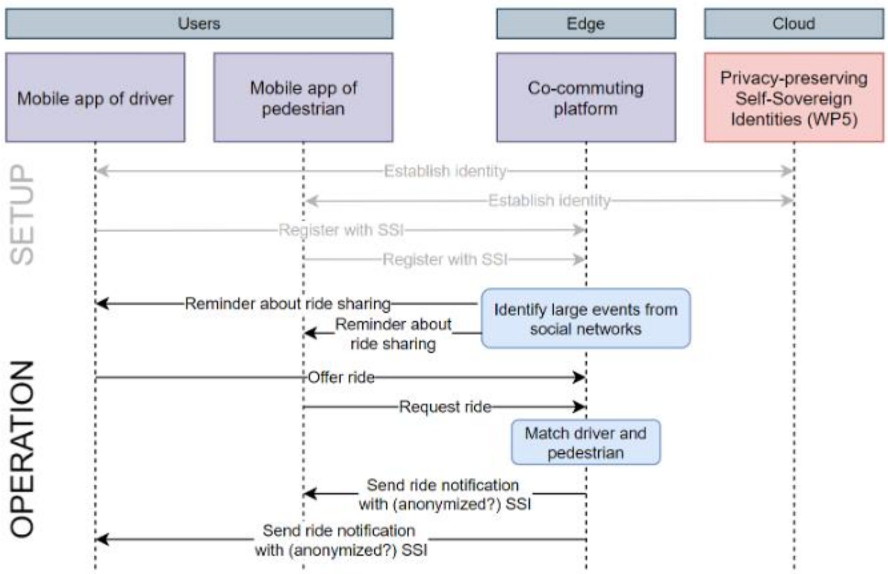


Figure 11: Sequence diagram of UC3.

3.1.3.6 Testing Scenarios

UC3 will be validated in the Jätkäsaari district, combining the integration of data pipelines from the Twitter feed, information from harbours and Helsinki Region infoshare.

Technical validation of the UC

- The key objective of UC3 is to demonstrate the use of various user-generated data sources in provision of co-commuting solutions, which is also reflected in KPI_SC_2.
- This use case also helps to validate the use of social networks as the data source for demand prediction.
- The Smart City living lab is used to aid the development of the SSI solutions to ensure they include the necessary properties to aid efficient development, deployment, and interfacing with other systems. This goal is then validated with the other living labs utilising these technologies.

Validation stages

- The components will first be validated individually in a lab environment during the development process.
- Then the whole solution will be validated in field conditions

User experience testing / validation

- UC3 user experience will be validated with selected questionnaires for the commuters. Targeted interviews are conducted to confirm the perceived effect on the business case.

Test scenario

The testing steps include:

1. Generate a twitter feed using volunteer smartphones.
2. Collect the twitter feed using features described in section 3.1.3.3.
3. Explore the extracted features to identify patterns, such as common topics, locations mentioned in the tweets, the date and time it was posted, and the contents of the tweet.
4. Test the accuracy of the prediction by comparing the predicted data with actual traffic data coming from HRI.
5. Test the integration of the analysed feed to 3rd party endpoint.

3.1.3.7 Execution timeline

The execution timeline of UC3, as already reported in D7.2, divided into multiple phases, is detailed in Table 10 below.

Table 10: Execution timeline of UC3.

Phase	Estimated start date	Estimated end date	Notes
Trial set-up and equipment procurement	M20	September 2022 (M24)	Completed. The Bills of materials (BOM) was collected to therefore start the implementation. BOM includes the data to be collected, devices to be used.
Initial implementation and validation	M25	M30	Initial twitter feed integration tested.

			Integration of data pipeline is in progress.
Final implementation and validation	M31	M36	Demonstrate the Twitter feed data flow and provide the proof of concept of such service to be feasible. Linking results between all 3 UCs. Potential business cases to be developed for sustainability.

3.1.3.8 Intermediate results

UC3 intermediate results are mainly related to collecting the bills of materials of the trial site setup, planning integration pipelines, preparing testing scenario and preparing the Urban data platform.

Data needed to perform the analysis of the social network feed was analyzed and specific features/objects of Twitter API v2 were selected. The updated data collection is described in the section 3.1.3.3.

In addition to that, initial analysis of Twitter API v2 features resulted in creation of the following objectives for investigation:

- Evaluate the effectiveness of different types of Twitter data features, such as geolocation data or user-generated content, for predicting traffic demand.
- Investigate the impact of different filtering methods on the accuracy of the prediction.
- Determine the optimal frequency of collecting Twitter data for real-time prediction.
- Evaluate whether Twitter data is detailed enough for co-commuting instruments to be enabled.
- Investigate the potential of using Twitter data for other use cases within the transportation industry, such as predicting public transit usage or identifying traffic congestion hotspots.

Additionally, as an intermediate result, business cases were developed demonstrating the benefits of UC3 in various scenarios. These cases, available in Section 3.1.3, showcase the potential impact of using data-driven insights to optimize traffic flows, parking allocation, and co-commuting solutions. These business cases provide tangible examples of how UC3 components can bring value to different contexts and can help stakeholders explore the potential of data-driven insights in transportation.

3.2 Smart Agriculture IoT Living Lab

3.2.1 UC4 – Crop diseases prediction, smart irrigation and precision aerial spraying

IoT-NGIN Use Case #4 "**Crop diseases prediction, smart irrigation and precision aerial spraying**" aims to address economic and food security challenges, which are strengthened by the sustaining decline of natural resources, but also by current political instabilities dominant agricultural countries.

Specifically, UC4 focuses on rationalizing the irrigation and spraying processes during plant growth, aiming to limit water and pesticides' usage to levels that are both sufficient for high-quality production. This is approached through innovations in 3 axes:

- Crop diseases prediction, based on ML models able to detect traces of crop diseases in leaves and thus identify specific areas in the field with diseased crops.
- Smart irrigation, which ensures that the irrigation process applies to the needs of the crop, considering the crop type, the soil type, as well as the current conditions of the crop, the soil and the environment.
- Precision aerial spraying, which is enabled by localized crop disease detection on the leaves.

For enabling this functionality, smart monitoring data collected by Synelixis' SynField®[2] [2] IoT devices are installed on the field, as well as aerial images captured by Unmanned Aerial Vehicles (UAVs) flying over the orchards. These data feed IoT-NGIN processes for the delivery of AI at the edge and cloud, Digital Twin services, improved interaction through AR and advanced access control, as well as cybersecurity for the AI and network part of the Smart Agriculture LL.

The following subsections present in detail the UC4 infrastructure, equipment, as well as the organization of pilot activities and results achieved so far.

3.2.1.1 Trial site description

The trial is conducted in a commercial vineyard at Peloponnese, Greece, in which a number of SynField nodes have been already installed. The pilot network topology has been already presented in D7.2 and is also included here in Figure 12.

[2] SynField nodes are the hardware (IoT) devices of Synelixis' SynField Smart Agriculture platform: https://www.synfield.gr

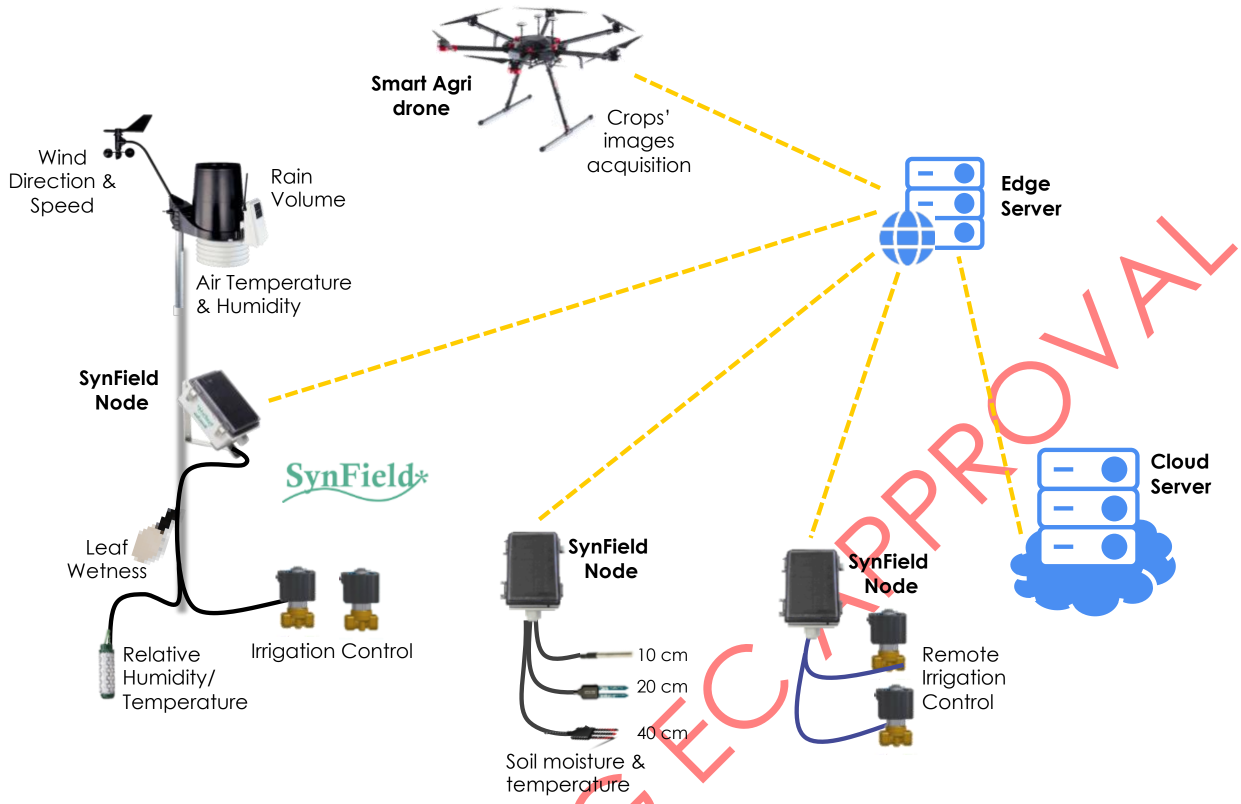


Figure 12: A graphic description of the trial site of UC4.

3.2.1.2 Required equipment

Table 11 below provides the detailed description of the required equipment for UC4, as already reported in D7.2, along with its type and status. As presented in the table,

Table 11: Detailed description of the required equipment for UC4.

Equipment	Description / Specifications	Type	Status
SynField Node	SynField Head Node (SF-HN-06) which is able to collect sensor measurements and transmit them to the edge or the cloud, as well as send commands to the actuators (e.g., for irrigation control)	IoT node integrating sensors and actuators	Deployed & integrated
UAV	The UAV solution that will fly over the orchard in order to capture aerial images and make crop disease detection on the fly. It is composed of the Smart Agri drone, the multi-spectral camera and the processing unit.	UAV	Prototype ready

	Smart Agri Drone	DJI Matrice 600 PRO, ideal for professional aerial photography with an extended flight time and a 5km long-range transmission, intelligent batteries, and maximum payload of 6kg.	drone	Integrated into UAV
	Multispectral camera	Parrot Sequoia, multi-band sensor designed for agriculture, featuring excellent precision, flexible integration, and small size and weight, compatible with the Smart Agri Drone.	Camera	Integrated into UAV
	Processing unit	NVIDIA Jetson Nano 4GB, a small, powerful computer which is able to run multiple neural networks in parallel for applications like image classification, which is desired for the crop disease prediction application.	Processing board	Integrated into UAV
Edge node		Edge node for running Digital Twin functionality and local ML model training. The edge node will host the client-side services of the IoT-NGIN framework.	Edge server	Available
Cloud node		Cloud resources used for permanent storage and ML model aggregation/training activities. Also, here the server-side services of IoT-NGIN will be deployed.	Cloud	Available within the Consortium

3.2.1.3 Data collection

The following dataset(s) have been identified as part of the use case:

- **Field sensor measurements:** data collected from SynField IoT platform and integrated sensors, including micro-climate data (air temperature, air humidity, wind direction, wind speed, rain volume, rain intensity) and soil data (soil moisture). This dataset will be used to calculate the crop growing degree days (ripening indicator).

- **Drone camera images:** RGB images collected from UAV's camera during its flight over the vineyard. The images are associated with time information and geospatial/location information provided by GPS.

3.2.1.4 Alignment with IoT-NGIN technologies

Within the scope of UC4, a number IoT-NGIN components will be validated in the Smart Agriculture LL. Table 12 below summarizes the context in which each component will be used in UC4, any adaptations needed and the planned deployment. Moreover, each component is associated with a set of requirements and Key Performance Indicators (KPIs), as defined in D1.1 [3]. This table updates the relevant information provided in D7.2.

Table 12: Alignment of UC4 with the relevant IoT-NGIN technologies.

WP2 – Secure edge-cloud execution framework	
Description	This framework will be used in order to enable secure deployment and operation of IoT-NGIN services. Indicatively, it will host the prediction models deployed on the agricultural drone.
Adaptation and fine-tuning	The Secure Execution framework will be integrated with the MLaaS platform, enabling secure deployment of ML models. As such, no adaptation is needed.
Deployment	Far edge node (UAV)
Related requirements	**REQ_SA1_NF01** – The IoT-NGIN platform must respect security and privacy requirements
Related KPIs	**KPI_SA1_01** – Reduction in the water needed for irrigation compared to manual irrigation = 20% **KPI_SA1_02** – Reduction in pesticides used for spraying compared to manual irrigation = 20% **KPI_SA1_03** – Increase in quantity of fruits harvested compared to manual irrigation and spraying >/=15%
WP3 – MLaaS framework	
Description	The MLaaS framework will be used to provide ML/AI services to the use case, related to ML model training for the crop disease prediction model and model serving for the image recognition model identifying SynField devices, as part of the access control mechanism.
Adaptation and fine-tuning	No adaptation of the MLaaS framework is needed for the purposes of UC4. The crop disease prediction application and AR application will consume the MLaaS services interacting with its API. The ML model for image recognition has to be stored in MLaaS and made available for model serving.
Deployment	A project-wide deployment can be used. The use case does not require a dedicated instance of the MLaaS framework.
Related requirements	**REQ_SA1_F05** – The platform should provide predictions of crop diseases, based on ML/FL performed over sensor measurements and drone images/videos **REQ_SA1_F06** – The platform should provide prediction accuracy probability **REQ_SA1_F09** – The platform should provide support for microclimate measurements sharing, respecting data sovereignty and privacy requirements **REQ_SA1_NF02** – IoT-NGIN should support High Availability features

Related KPIs	**KPI_SA1_01** – Reduction in the water needed for irrigation compared to manual irrigation = 20% **KPI_SA1_02** – Reduction in pesticides used for spraying compared to manual irrigation = 20% **KPI_SA1_03** – Increase in quantity of fruits harvested compared to manual irrigation and spraying >/=15%
WP3 – Privacy-preserving Federated Machine Learning (PPFL)	
Description	This framework will be used in order to allow training the ML models at multiple edge nodes in a federated-distributed manner. The locally trained models will be aggregated at the cloud server. The training refers to crop disease prediction models.
Adaptation and fine-tuning	The FL framework will be configured to work with a number of "participants" (clients) equal to the number of edge nodes used in UC4.
Deployment	Multi-layer, including edge and cloud resources, as described above.
Related requirements	**REQ_SA1_NF01** – The IoT-NGIN platform must respect security and privacy requirements
Related KPIs	**KPI_SA1_01** – Reduction in the water needed for irrigation compared to manual irrigation = 20% **KPI_SA1_02** – Reduction in pesticides used for spraying compared to manual irrigation = 20% **KPI_SA1_03** – Increase in quantity of fruits harvested compared to manual irrigation and spraying >/=15%
WP3 – Polyglot Model Sharing component	
Description	This component will allow serving the ML image recognition model to AR application running e.g. on the Smart Farmer's mobile phone
Adaptation and fine-tuning	No adaptation need identified at this point.
Deployment	As part of the MLaaS platform
Related requirements	**REQ_SA1_F07** – The platform should provide automated irrigation and aerial spraying based on crop disease prediction
Related KPIs	**KPI_SA1_01** – Reduction in the water needed for irrigation compared to manual irrigation = 20%

	KPI_SA1_03 – Increase in quantity of fruits harvested compared to manual irrigation and spraying >/=15%
WP4 – IoT device discovery (IDD) (Computer Vision)	
Description	The Computer Vision (CV)-based recognition parts of the component will allow identification of SynField nodes needed for the delivery of AR services through an application on the user's mobile device. This identification will support one of the criteria set for access management to the SynField monitoring data and actuation.
Adaptation and fine-tuning	CV models have been trained and validated over SynField node images and images.
Deployment	Cloud (on the MLaaS platform)
Related requirements	**REQ_SA1_F01** – The platform should provide access to measurements. **REQ_SA1_F02** – The platform should provide options to manage/view sensors/devices. **REQ_SA1_F16** – The platform should allow the user to execute manually the recommended irrigation or spraying plans.
Related KPIs	**KPI_SA1_01** – Reduction in the water needed for irrigation compared to manual irrigation = 20%
WP4 – IoT device indexing (IDI)	
Description	This module will support the Digital Twin of SynField devices and UAVs. Within UC4, it will allow accessing data collected by SynField devices and UAVs, as well as any information accompanying these devices.
Adaptation and fine-tuning	No adaptation need identified at this point.
Deployment	Edge
Related requirements	**REQ_SA1_F01** – The platform should provide access to measurements. **REQ_SA1_F02** – The platform should provide options to manage/view sensors/devices. **REQ_SA1_F08** – The platform should provide access to irrigation and aerial spraying data. **REQ_SA1_F10** – The platform must monitor the weather and plant conditions. **REQ_SA1_F11** – The platform must be able to retrieve plant photos via drones. **REQ_SA1_NF04** – The UC4 Application of IoT-NGIN should be vendor-independent.

	REQ_SA1_NF05 – The UC4 Application of IoT-NGIN should be scalable in terms of adding/removing devices or device types and integrating hundreds of devices.
Related KPIs	**KPI_SA1_04** – Number of sensors tested for connectivity with IoT-NGIN Smart Agriculture app > 9
WP4 – IoT device access control (IDAC)	
Description	This module will also support the Digital Twin functionality, providing access management for collected data and actuations made available through the IoT Device Indexing.
Adaptation and fine-tuning	No adaptation need identified.
Deployment	Edge
Related requirements	**REQ_SA1_F08** – The platform should provide access to irrigation and aerial spraying data. **REQ_SA1_F14** – The platform should allow the eligible user to provide irrigation and spraying suggestions. **REQ_SA1_F15** – The platform should allow the user to view the irrigation and spraying suggestions. **REQ_SA1_NF01** – The IoT-NGIN platform must respect security and privacy requirements.
Related KPIs	**KPI_SA1_01** – Reduction in the water needed for irrigation compared to manual irrigation = 20% **KPI_SA1_02** – Reduction in pesticides used for spraying compared to manual irrigation = 20% **KPI_SA1_03** – Increase in quantity of fruits harvested compared to manual irrigation and spraying >/=15%
WP4 – IoT AR toolkit	
Description	The user will be able to access monitoring data of SynField nodes via scanning them and using Augmented Reality (AR) illustration of buttons on the SynField device appearing on the user's device.
Adaptation and fine-tuning	No adaptation need identified at this point.
Deployment	IoT
Related requirements	**REQ_SA1_F07** – The platform should provide automated irrigation and aerial spraying based on crop disease prediction.

	REQ_SA1_NF03 – The UC4 Application of IoT-NGIN should support UX features
Related KPIs	**KPI_SA1_01** – Reduction in the water needed for irrigation compared to manual irrigation = 20%
WP5 – GAN-based dataset generation	
Description	UC4 will use synthetic datasets for validating the attack detection models, which will identify network/data poisoning level attacks at the Smart Agriculture LL infrastructure.
Adaptation and fine-tuning	The Generative Adversarial Network (GAN) based generator model will be trained using crop images/network logs of the SynField infrastructure/network.
Deployment	Edge
Related requirements	**REQ_SA1_NF01** – The IoT-NGIN platform must respect security and privacy requirements
Related KPIs	**KPI_SA1_01** – Reduction in the water needed for irrigation compared to manual irrigation = 20% **KPI_SA1_02** – Reduction in pesticides used for spraying compared to manual irrigation = 20% **KPI_SA1_03** – Increase in quantity of fruits harvested compared to manual irrigation and spraying >/=15%
WP5 – Malicious Attack Detector (MAD)	
Description	This component will identify network-level attacks on the SynField system.
Adaptation and fine-tuning	The ML model should have been trained over network datasets, similar to the network of the Smart Agriculture.
Deployment	Edge
Related requirements	**REQ_SA1_NF01** – The IoT-NGIN platform must respect security and privacy requirements.
Related KPIs	**KPI_SA1_01** – Reduction in the water needed for irrigation compared to manual irrigation = 20% **KPI_SA1_02** – Reduction in pesticides used for spraying compared to manual irrigation = 20% **KPI_SA1_03** – Increase in quantity of fruits harvested compared to manual irrigation and spraying >/=15%
WP5 – IoT vulnerability crawler (IVC)	

Description	This component will identify potential vulnerabilities in SynField nodes, the UAV and the edge nodes.
Adaptation and fine-tuning	No adaptation need identified at this point. The component has to be deployed within the Smart Agriculture LL network.
Deployment	Edge/Cloud
Related requirements	**REQ_SA1_NF01** – The IoT-NGIN platform must respect security and privacy requirements.
Related KPIs	**KPI_SA1_01** – Reduction in the water needed for irrigation compared to manual irrigation = 20% **KPI_SA1_02** – Reduction in pesticides used for spraying compared to manual irrigation = 20% **KPI_SA1_03** – Increase in quantity of fruits harvested compared to manual irrigation and spraying >/=15%
WP5 – Moving Target Defences (MTDs) network of honeypots	
Description	This module will be used to dynamically set up networks of honeypots in a restricted area, in a subdomain of SynField.
Adaptation and fine-tuning	No adaptation need identified at this point.
Deployment	Edge/Cloud
Related requirements	**REQ_SA1_NF01** – The IoT-NGIN platform must respect security and privacy requirements.
Related KPIs	**KPI_SA1_01** – Reduction in the water needed for irrigation compared to manual irrigation = 20% **KPI_SA1_02** – Reduction in pesticides used for spraying compared to manual irrigation = 20% **KPI_SA1_03** – Increase in quantity of fruits harvested compared to manual irrigation and spraying >/=15%

3.2.1.5 Testing Scenarios

For validating the IoT-NGIN efficiency towards providing added value in Smart Agriculture in relation to smart irrigation, crop disease prediction and aerial spraying, three test scenarios have been defined and are being implemented in the LL with the use of the IoT-NGIN tools. The test scenarios are described in the following subsections.

3.2.1.5.1 Smart Irrigation and AR-based actuation

Table 13: UC4 Test 1 “Smart irrigation and AR-based actuation”.

Test 1: Smart irrigation and AR-based actuation	
Objective	The objective is to improve the effectiveness of the irrigation process, ensuring appropriate use of water resources as well as high grape quality and yield. At the same time, we aim at minimizing faults or malicious activities in irrigation management which may affect the yield, as well as enhance the smart farmer’s interaction with the smart agriculture IoT devices.
Components	• IoT Device Discovery (IDD) • IoT Device Indexing (IDI) • IoT Device Access Control • AR module • MLaaS • Smart Agri Mobile app • SynField platform (by Synelixis)
Features to be tested	• Development of Smart Farm’s Digital Twin, realized through the IDI component • Data acquisition from SynField nodes Digital Twin • ML-based Image recognition service • Pervasive security in the context of ambient intelligence, granting device access based on user credentials, location information and device recognition • AR visualization of SynField node info on user’s mobile device • AR-based actuation, which controls the node’s irrigation modules
Requirements addressed	REQ_SA1_F01, REQ_SA1_F02, REQ_SA1_F04, REQ_SA1_F08, REQ_SA1_F10 REQ_SA1_F12, REQ_SA1_F13, REQ_SA1_F14, REQ_SA1_F15, REQ_SA1_F16, REQ_SA1_NF01, REQ_SA1_NF03, REQ_SA1_NF04, REQ_SA1_NF05
Test setup	The IoT-NGIN framework should be deployed and become functional. A set of SynField nodes is installed and configured to communicate with the edge node and the SynField platform. The SynField platform must be integrated with IDI, in order to inject sensor measurements, but also receive actuation commands. The Smart Agri App is installed on the user’s device and the user is eligible to access data of at least one of the installed SynField nodes, protected by the IDAC module. The Computer Vision (CV)-based image recognition module of the IDD component should be deployed and accessible via the MLaaS platform, based on already trained ML model, able to recognize SynField nodes. Each SynField node carries a QR code, which provides the node’s id information.
Steps	Baseline Case #1 – static 1. The user sets a static irrigation plan 2. The user accesses the monitoring data via the SynField nodes’ Digital Twin Case #2 – semi-automatic with and without 3-step access control

	1. The user sets the irrigation in semi-automatic mode, allowing the Smart Farmer to receive recommendations on the irrigation which have to be manually applied.
	2. The user receives a recommendation for activating irrigation.
	3. A. The user approaches a wrong SynField node.
	    a. If 3-step access control is not active, the user activates directly the node.
	    b. If 3-step access control is active,
	        i. the users scans the node's QR code through the Smart Agri app and streams SynField node's images through the Smart Agri app, in order to get access to the node's data
	        ii. the user is notified that this is not the correct node and is not granted access to actuate the device.
	B. The user approaches a correct SynField node.
	    a. If 3-step access control is not active, the user directly activates the node.
	    b. If 3-step access control is active,
	        i. the users scan the node's QR code through the Smart Agri app and streams SynField node's images through the Smart Agri app, in order to get access to the node data
	        ii. the user is granted access to the AR functionality and visualizes information about the SynField node, as well as its monitored data and calculated predictions via AR on their device. The user actuates the device.
	4. The user accesses the monitoring data via the SynField nodes' Digital Twin.
	
	Case#3 - Automatic
	1. The user sets the automatic irrigation plan, applying intelligence gained over the monitored SynField data.
	2. The user accesses the monitoring data via the SynField nodes' Digital Twin.

KPIs	KPI_SA1_01, KPI_SA1_03, KPI_SA1_04
Innovations brought by IoT-NGIN	• Monitoring data are available on IoT-NGIN's DT, protected by the Access Control. • Any calculation and analytics takes place at the edge – without need to send data to the cloud. • Access to SynField devices is protected and controlled through multiple criteria, selected for the use case. • Enhanced interaction with SynField devices, could be even easier via future user devices (e.g., wearables).

3.2.1.5.2 ML-based crop disease prediction and aerial spraying

Table 14: UC4 Test 2 "ML-based crop disease prediction and aerial spraying".

Test 2: ML-based crop disease prediction and aerial spraying	
Objective	The objective of this test is to maintain crops' health in a sustainable way, both from the environmental and the economic perspective, through the ML-based crop disease prediction functionality of the IoT-NGIN UC4 application.
Components	• IoT Device Indexing • IoT Device Access Control • MLaaS • Privacy-preserving Federated Learning • Secure Execution Framework

Features to be tested	• ML-based crop disease prediction service development
	  ○ Training
	  ○ Deployment
	  ○ Inference
	• Smart Farm' Digital Twin
	• Access control

Requirements addressed	REQ_SA1_F01, REQ_SA1_F02, REQ_SA1_F03, REQ_SA1_F04, REQ_SA1_F05, REQ_SA1_F06, REQ_SA1_F07, REQ_SA1_F08, REQ_SA1_F09, REQ_SA1_F10, REQ_SA1_F11, REQ_SA1_F12, REQ_SA1_F13, REQ_SA1_F14, REQ_SA1_F15, REQ_SA1_F16, REQ_SA1_NF01, REQ_SA1_NF04, REQ_SA1_NF05
Test setup	The IoT-NGIN framework should be installed and functional. The IoT-NGIN-enabled UAV should be prepared (integrating camera, processor, and batteries). The user should be provided with permissions to access and control UAVs. The UAV is integrated with its Digital Twin at the edge device, realized via IDI and IDAC and keeping its data acquired on a regular basis, but it can also receive control commands through it, e.g., for ML model updates.
Steps	1. The user triggers the training of a new ML model for crop disease prediction. 2. The user triggers the deployment of the new model into the UAV, which is executed via its Digital Twin. 3. The user commands a UAV flight over the vineyard and monitors its movement. 4. The UAV captures images of the vineyard and performs on-device inference. 5. After the UAV flight has been completed, the user accesses the UAV data and predictions.
KPIs	KPI_SA1_02, KPI_SA1_03, KPI_SA1_04
IoT-NGIN innovations	• UAV data and predictions are available on IoT-NGIN's DT, protected by the Access Control. • Efficient use of resources and minimization of communication costs, as ML inference takes place at the far edge node (UAV), while model training happens at the edge. No need to send data to the cloud for edge analytics.

	• Continuous enhancements to the prediction efficiency are possible, through a simple and automated process for ML model training and deployment on the UAV.

3.2.1.5.3 Smart agriculture cybersecurity

Table 15: UC4 Test 3 "Smart agriculture cybersecurity".

Test 3: Smart agriculture cybersecurity	
Objective	The objective of this test is to validate cybersecurity in IoT-NGIN-induced tools in the Smart Agriculture LL, effectively protecting the UC4 operation from IoT cyber-attacks which could compromise the effectiveness of the AI models.
Components	• IoT Device Indexing • IoT Device Access Control • GAN-based Dataset Generator • Malicious Attack Detection • IoT Vulnerability Crawler • MTD Network of Honeypots
Features to be tested	• Vulnerability scanning of SynField devices • Malicious attack detection • Malicious attack pattern investigation for the Smart Agriculture domain • Smart Farm Digital Twin • Access control
Requirements addressed	REQ_SA1_F01, REQ_SA1_F12, REQ_SA1_NF01, REQ_SA1_NF02
Test setup	The IoT-NGIN framework should be installed and become functional. The Digital Twin services for the SynField IoT devices should be available at the edge node. The user should be provided with permissions to access monitored data and cybersecurity reports generated by IoT-NGIN. Network access is provided to allow vulnerability scanning for networked devices, i.e., SynField IoT nodes and UAVs. Network data from the Smart Agriculture LL should be captured and made available to the IoT-NGIN cybersecurity modules. The malicious attack detection modules are deployed and operate over such data. A controlled, isolated subnetwork under the Smart Agriculture's LL domain should be available to host threat monitoring via the MTD Network of Honeypots.
Steps	1. The user accesses the vulnerability scanning outcomes for the Smart Agriculture LL. 2. The user approves the setup of a new honeypot based on the identified vulnerabilities. 3. The user accesses the malicious attack detection outcomes. a. Depending on the detected attack, the attacked node data are excluded from the next training cycle. 4. The administrator resets the attacked network/node.
KPIs	KPI_SA1_01, KPI_SA1_02, KPI_SA1_03
IoT-NGIN innovations	• Edge-native cybersecurity protection for Smart Agricultural AI services, which has received low attention so far

	• Systematic threat prevention, monitoring, detection and mitigation in Smart Agriculture IoT systems

3.2.1.6 Use case sequence diagrams

The updated use case diagrams for UC4 are presented in the following. Figure 13 presents the sequence diagram for smart irrigation and AR-based actuation.

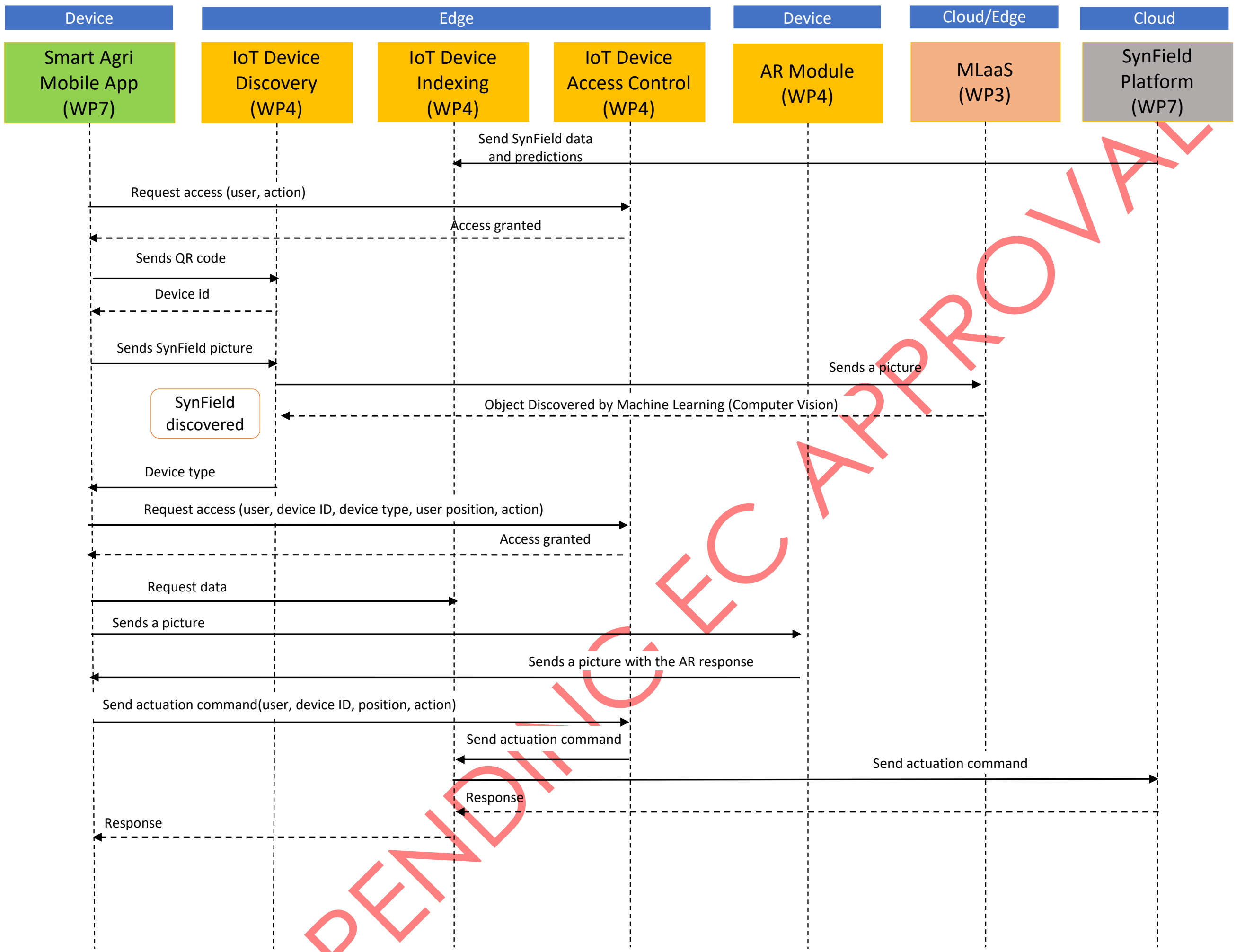


Figure 13: Sequence diagram for smart irrigation and AR-based actuation.

Figure 14 presents the sequence of operations for the second testing scenario "ML-based crop disease prediction". As indicated in the figure, the ML model training, deployment and inference processes are described, indicating the connection of cloud, edge and IoT for ML operations through the UAV's digital twin. The update of the ML model for crop disease prediction running on the UAV is conducted through the IDI and IDAC components, supporting the UAV's digital twin.

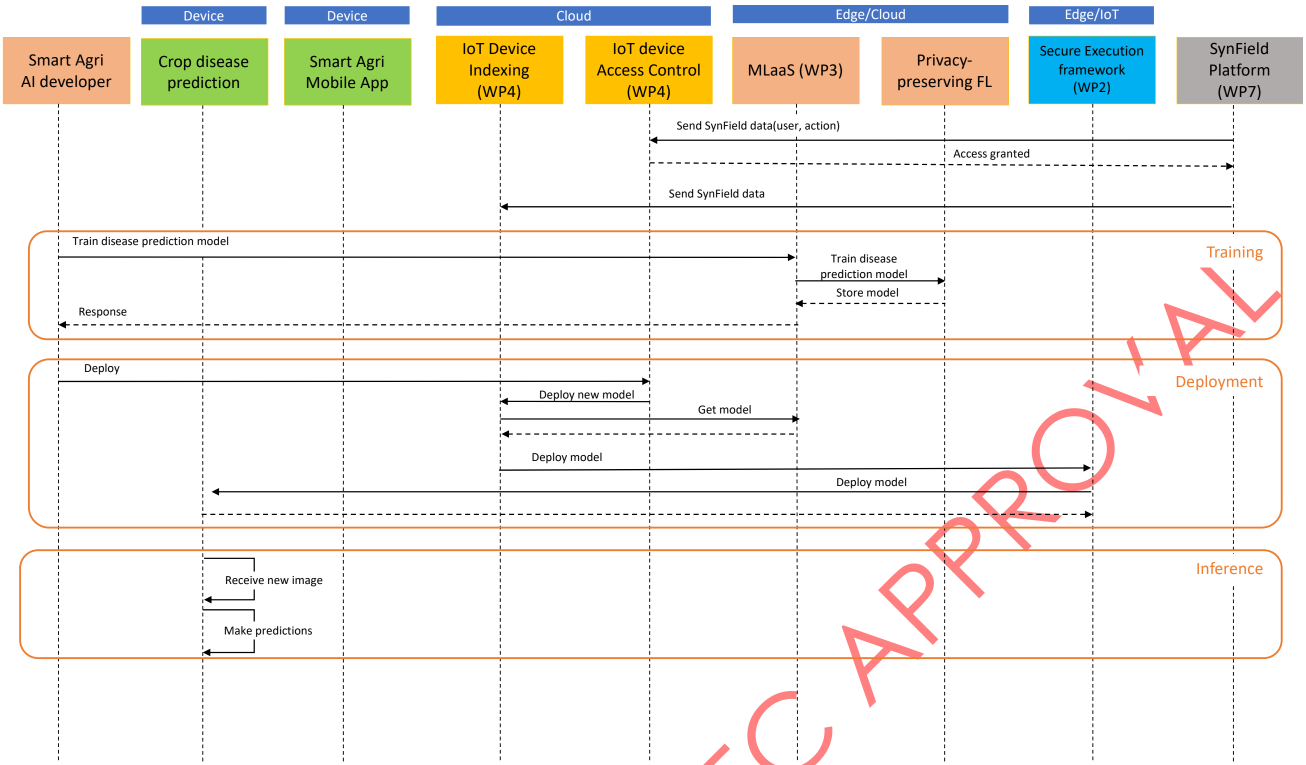


Figure 14: Sequence diagram for ML-based crop disease prediction.

Figure 15 presents the sequence diagram for the cybersecurity testing scenario. Here, we see the role of the GAN based Dataset Generator for the creation of poisoned datasets, as well as the operation of MAD for the detection of induced attacks. Also, the cooperation is IVC with the MTD honeypots framework for vulnerability scanning and threat monitoring is also presented in this sequence diagram.

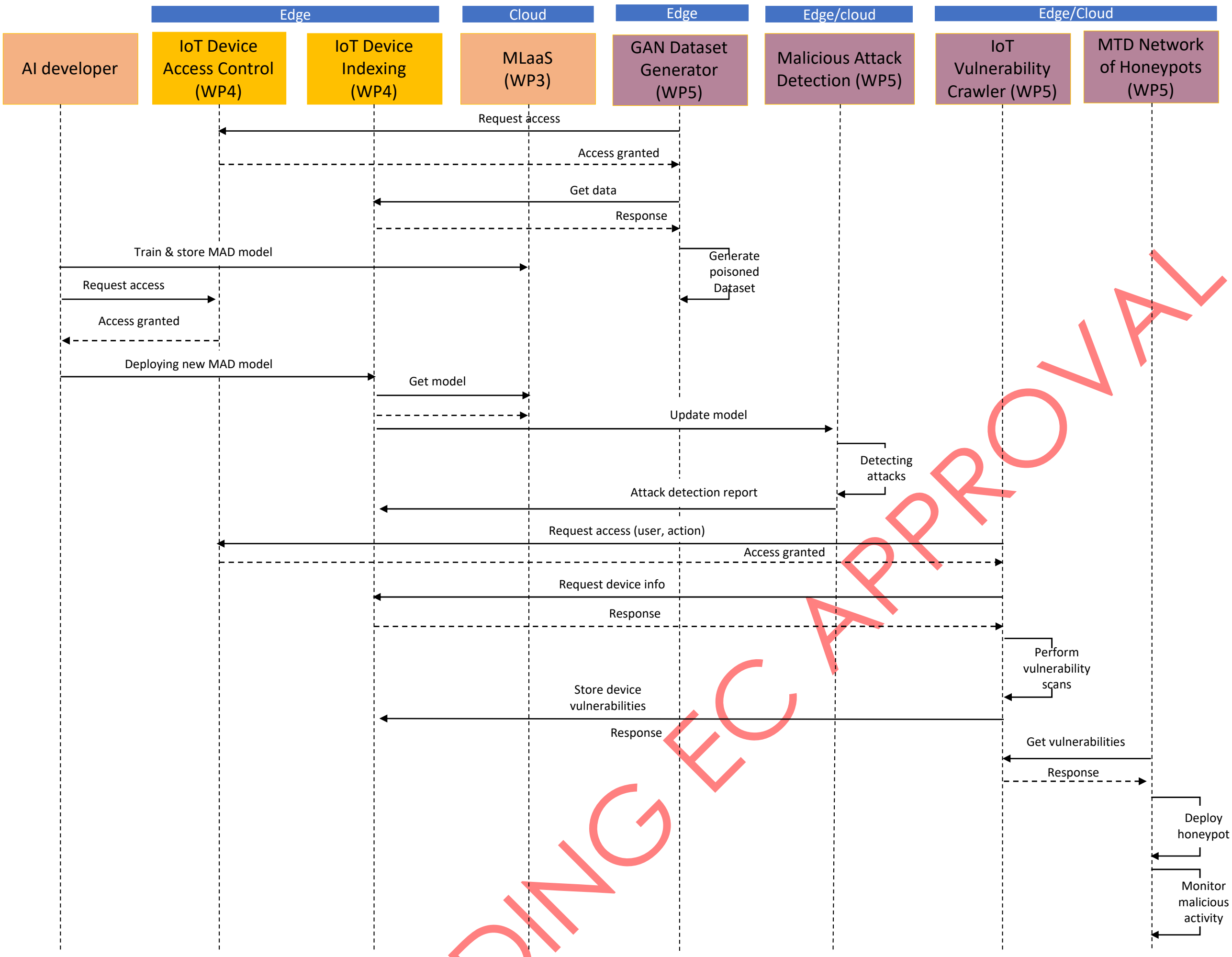


Figure 15: Sequence diagram for IoT-NGIN cybersecurity features in Smart Agriculture LL.

3.2.1.7 Execution timeline

The pilot activities are conducted, according to the plan presented in Table 16, which illustrates that the second wave of the LL activities, concerning the deployment and functional validation of the IoT-NGIN tools, has been completed. The LL validation activities, against the three defined scenarios, will be running until the end of the project.

Table 16: Execution timeline of UC4.

Phase	Estimated start date	Estimated end date	Notes
Trial set-up and equipment procurement	*M5*	*M18*	Completed
Initial implementation and validation	*M19*	M30	Completed

Final implementation and validation	*M31*	*M36*	Validation mainly on the basis of the final integrated prototype

3.2.1.8 Intermediate results

The pilot activities for UC4 so far have been focused on:

- Data collection through integration of IoT-NGIN's digital twinning functionality with the SynField platform, as well as the UAV
- UAV prototype validation, able to support on-device intelligence
- IoT-NGIN framework deployment and validation on the Smart Agriculture LL

For the needs of the IoT-NGIN validation, IoT-NGIN tools need to be installed in the Smart Agriculture LL. Following the integration & deployment approach of the project, a K8S installation has been made available in the LL facilities and IoT-NGIN tools have been deployed in a dedicated namespace '*iot-ngin*'. Specifically, the following components have been deployed on the Smart Agriculture LL:

- IoT Device Indexing
- IoT Device Access Control
- Privacy-preserving FL API
- IoT Vulnerability Crawler
- Malicious Attack Detector

Moreover, the MLaaS platform deployment in OneLab is used for the delivery of ML services, integrating the CV-based image recognition part of the IoT Device Discovery component, as well as the Model Sharing component.

In the following, the LL activities for each of the test scenarios under UC4 are briefly presented.

3.2.1.8.1 Smart Irrigation and AR-based actuation

For the execution of this scenario, a set of SynField IoT nodes deployed on the field has been used, in order to collect monitored data from the sensors attached to them. Specifically, eight (8) SynField nodes are integrated in IoT-NGIN pilots, which carry sensors measuring the following properties:

- Air humidity
- Air temperature
- Rain
- Wind direction
- Wind speed
- Soil moisture

These data are used for disease prediction, based on the environmental conditions.

The SynField data are available through the SynField platform dashboard, as depicted in Figure 16, while the current measurements of a single node are visualized in the SynField dashboard as shown in Figure 18. The full list of SynField nodes that are involved in UC4 are depicted on the map of Figure 17 through the same dashboard.

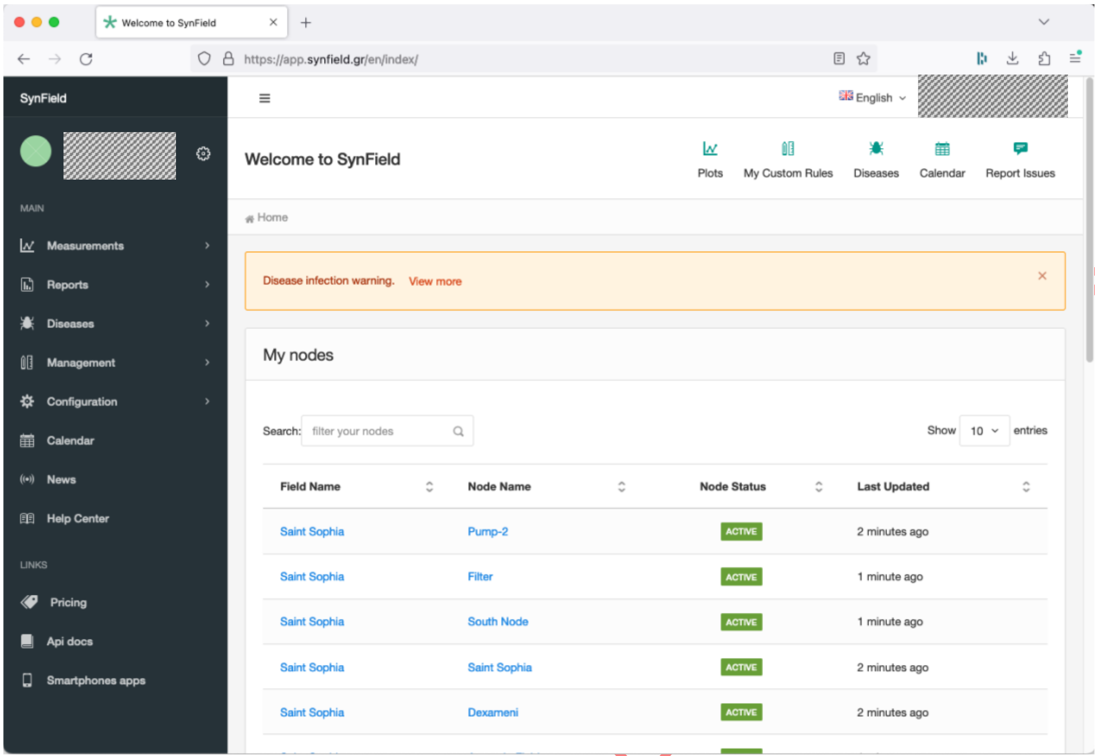


Figure 16: SynField nodes through the SynField® platform dashboard.

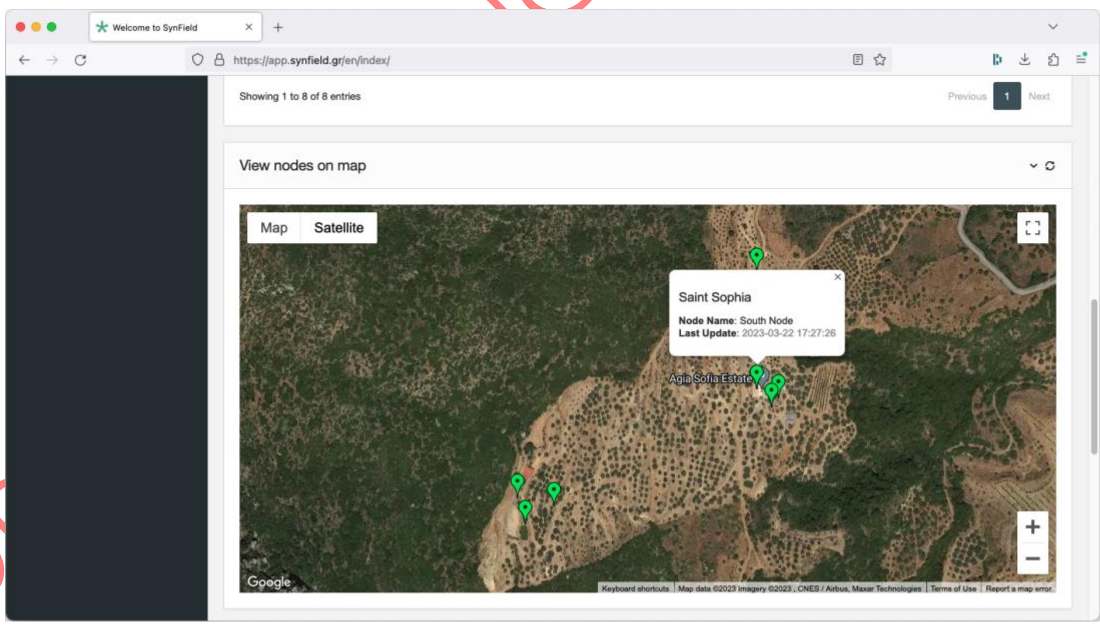


Figure 17: The SynField nodes used in UC4 on the map in the SynField® platform dashboard.

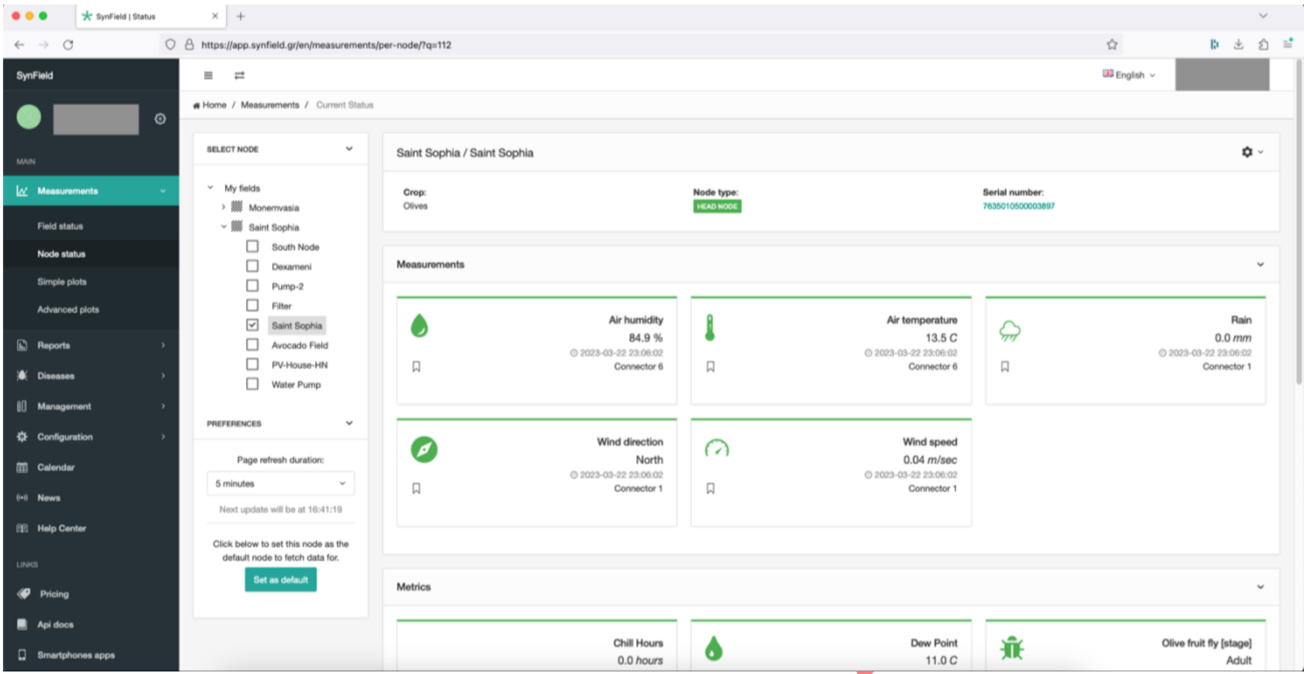


Figure 18: Latest sensor measurements of a SynField node visualized in the SynField dashboard.

For the purposes of UC4, we need to collect the measurements of these SynField nodes into IoT-NGIN. This will allow to create the Digital Twin of the SynField devices at the edge node of the trial and then apply edge analytics on them for the purposes of smart irrigation.

Therefore, the IoT Device Indexing and Access Control components of IoT-NGIN, which deliver DT functionality, have been deployed at the LL facilities and populated with the SynField data. Figure 19 and Figure 20 present via KubeView [4] these components' deployments in the K8S cluster of the Smart Agriculture LL.

Figure 19: IoT Device Access Control deployment on Smart Agriculture LL K8S cluster.

Figure 20: IoT Device Indexing deployment on Smart Agriculture LL K8S cluster.

Both historical and real-time data can be accessed through simple REST API calls to the Digital Twin, supported by IDI and IDAC. Specifically, Figure 21 depicts an indicative request for accessing latest measurements of a single SynField device, the same as the one presented in Figure 18 through the SynField dashboard. The sensing id and values correspond to the sensors already shown in the dashboard and we can see that the sensed values are the same in both cases.

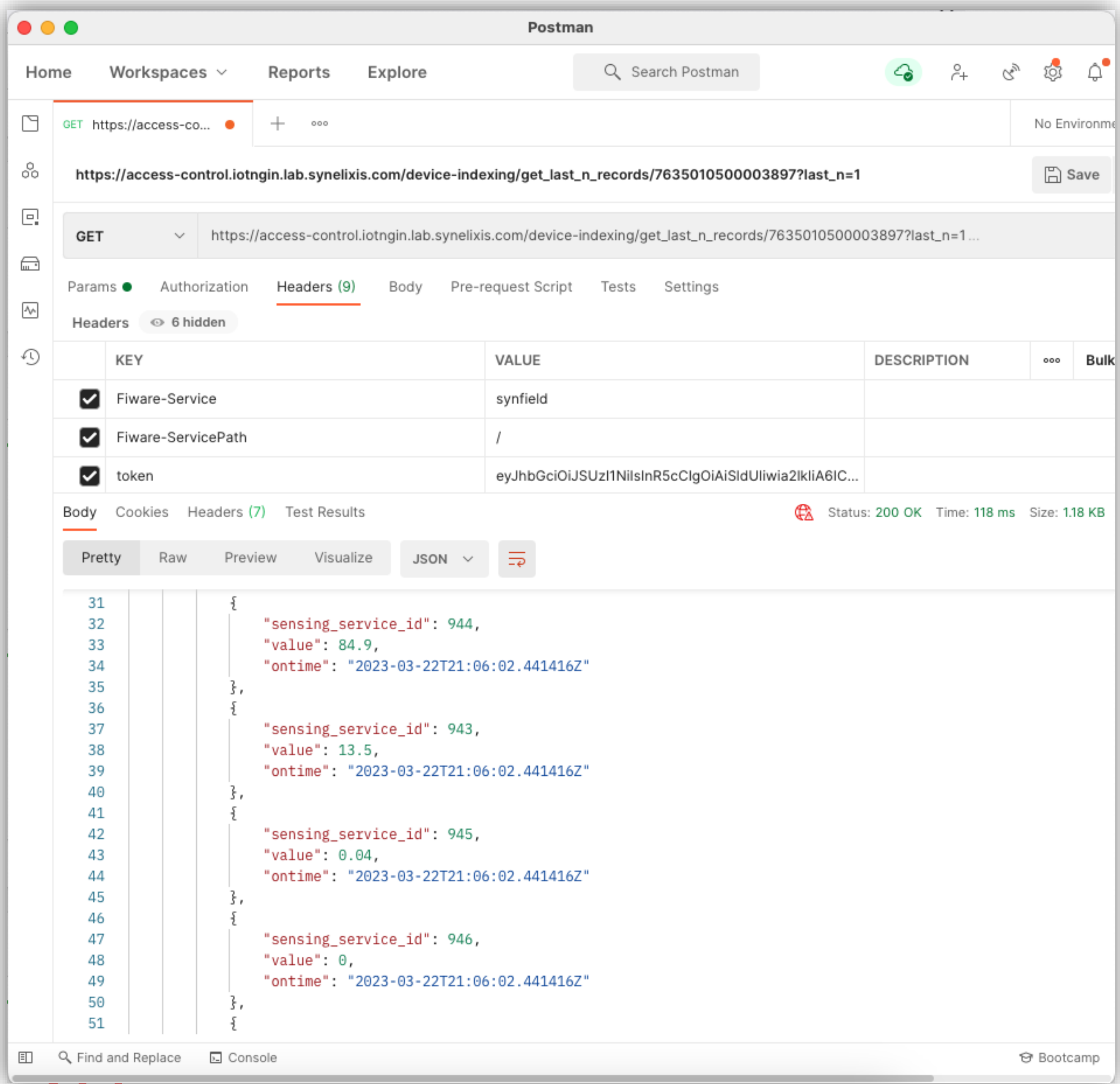


Figure 21: Latest sensor measurements of a SynField node provided via its Digital Twin.

Similarly, Figure 22 depicts the case for accessing historical data of the same device for a specified period, made available through its digital twin.

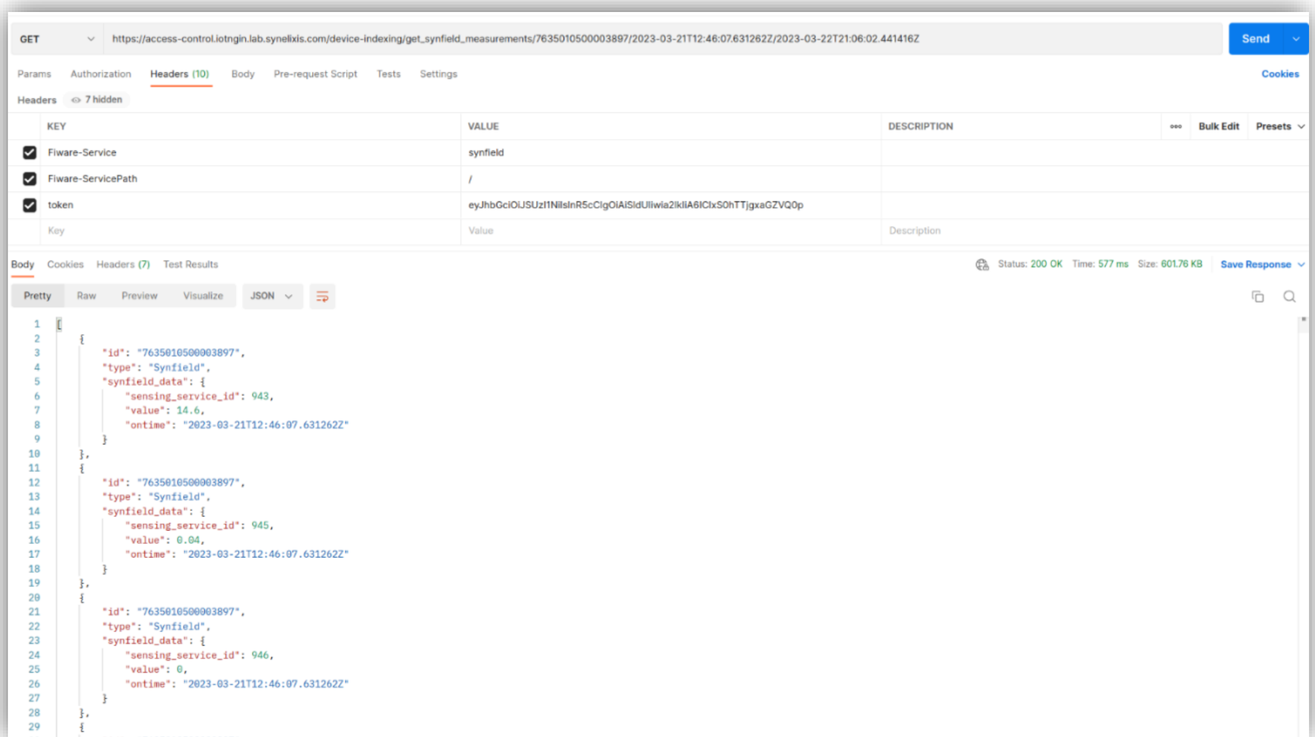


Figure 22: Historical measurements of a single SynField node for a given period, provided through its Digital Twin.

3.2.1.8.1.1 ML-based crop disease prediction and aerial spraying

The pilot activities in under this scenario include:

- Experimenting in real environment with the ML model for crop disease prediction and applying enhancements. A first version of the model has been already described in D6.2 [5].
- Integration of the model in the pilot equipment. Specifically, the crop disease prediction service has been deployed on a Jetson Nano board, receiving input from the multispectral camera.
- Integration of the Jetson Nano board and the camera on the drone
- Experiments with UAV flights, collect vineyard images and verify the operation of the crop disease prediction service, as well as the logging infrastructure for test results' evaluation.

3.2.1.8.1.2 Experiments with the crop disease prediction service

Within the framework of the crop disease prediction service, an ML model for the detection of Esca-diseased grapevine plants has been developed and described in D6.2 [5]. While this model has been effective in controlled laboratory tests, the tests in realistic conditions indicated that the model would anyway classify any input image as healthy or diseased, even if the input image depicted no plants. Although this is reasonable from the perspective of ML development, in practice it causes confusion as per the conclusion on diseased areas in the field, as well as the recommendations for spraying.

Thus, the model has been slightly enhanced to remediate classification of images without grapevine plants. In the following, the re-training process of the model is described, providing details on the dataset selection, the architecture of the model, and the model's performance. Next, the integration of the model on the pilot equipment, as well as the inference process on it are briefly presented.

3.2.1.8.1.3 Model training

Dataset Selection

The dataset used for developing the initial model is the ESCA dataset [6], as it provides a wide set of images which are appropriate for developing an accurate model. The dataset provides a set of RGB (Red, Green, Blue) images of grapevine leaves. The images consist of two different classes. The first class belongs to the grapevine leaves that are affected by the Esca disease, while the second one describes healthy grapevine leaves. The dataset contains 1770 images in total, where 888 belong to the 'esca' class and the rest 882 images belong to the 'healthy' class. To improve the generalization of the model and make it more robust, a new class called the 'unknown' class has been added to the dataset. In particular, the "Nature Image Classification" dataset [7] has been used for the 'unknown' class, which consists of 6 different classes of natural scenes, including buildings, forest, glacier, mountain, sea, and street. The addition of the 'unknown' class improves the model's ability to handle real-world scenarios and better classify captured images, when it encounters an object or scene that does not belong to any of the two classes of grapevine leaves. Furthermore, to balance the dataset, some samples from each class of the "Nature Image Classification" dataset were removed, resulting in a total of 888 samples. Finally, data augmentation techniques have been applied, as proposed in [6], in order to increase the size of the dataset. Therefore, after the data augmentation process, the final size of the dataset is 12,432 for the 'esca' class, 12,348 for the 'healthy' class and 12,432 for the 'unknown' class.

Model Architecture

The architecture is the same as for the initial model described in D6.2. A simple CNN model has been used for training. The CNN architecture consists of 5 Convolutional 2D layers followed by ReLu activation function and 5 2D Max Pooling with a 2 x 2 pool size. In the final stage, a flatten, two dense layers, with ReLu and softmax activation functions have been attached respectively to classify the provided input training images.

Model training and performance

Before the training process the data have been split to use 80% of the dataset for training, 10% for validation, and the rest 10% for testing. Table 17 describes the batch size and the dataset size, providing the distribution of the dataset for each scope in absolute numbers. The model has been trained on the NVIDIA TITAN X Graphics Processing Unit (GPU) with 32 GB Random-Access Memory (RAM) and the duration of the model training lasted about an hour. The number of epochs used to train the network is 50 for all the experiments with 32 batch size.

Table 17: Information about batch size and split of dataset.

Epochs	Batch size	Total size	Training dataset	Testing dataset	Validation dataset

50	32	37.212	29.769	3.721	3.722

The accuracy of the model reached 96.67% on the test data. Figure 23 visualizes the performance of the model during every 5 epochs.

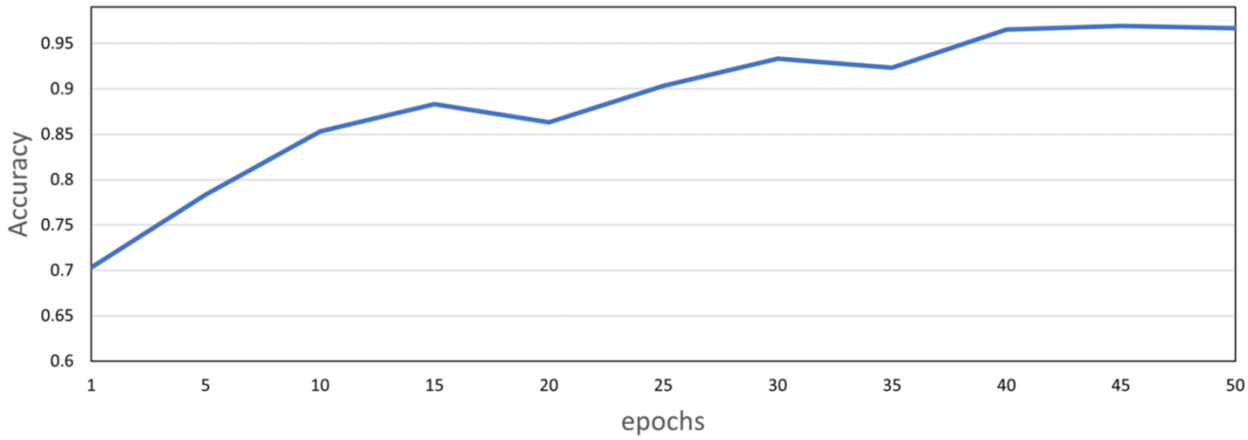


Figure 23: Performance of the model.

Upon completion of the training process, the inference service intended for the UAV has been updated with the trained model. This service receives captured images as input, which have first undergone preprocessing, such as conversion to tensors, before the trained model is loaded to predict if the grapevine leaves in the image are infected with Esca, are healthy, or belong to the ‘unknown’ class. An illustration of this process is presented in Figure 24, where the inference service accurately labels each image that appears in the input folder.

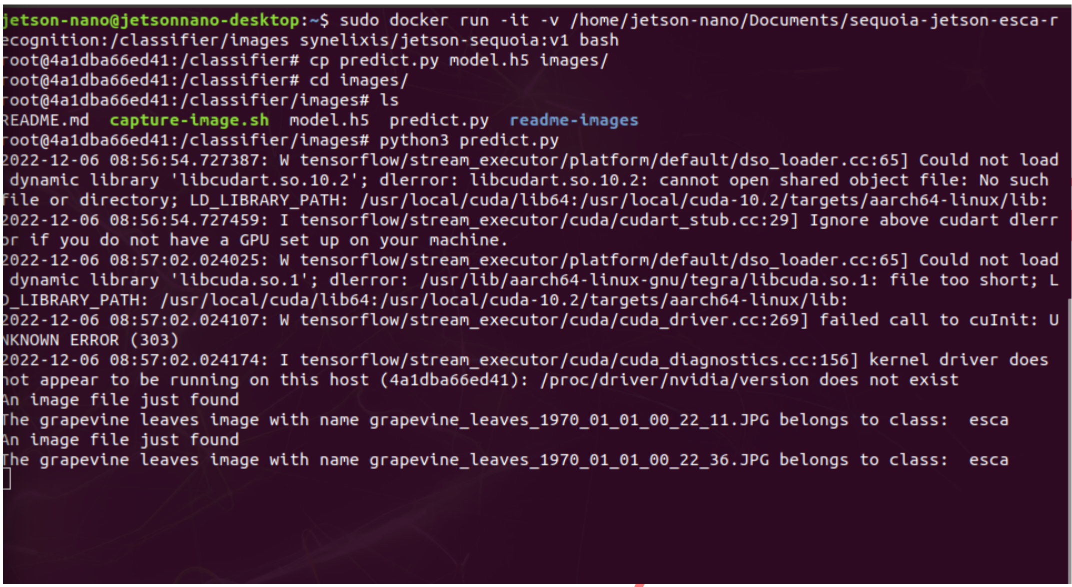


Figure 24: The prediction outcome of the crop disease prediction service.

3.2.1.8.1.4 Integration of Sequoia with Jetson-Nano

This section describes the integration of the multispectral camera with the processing unit, specifically of the Sequoia camera by Parrot with the NVIDIA Jetson Nano board. The integration of these two components enables capturing high-quality images and performing inference using the deep learning model presented in the previous subsection.

Sequoia is a multispectral camera produced by Parrot, appropriate for use in agriculture and other industries. The Sequoia camera gives output images in four spectral bands (green, red, red edge, and near-infrared) allowing to provide insights into plant health, growth, and stress levels. Also, it captures RGB images with its 16-megapixel RGB sensor. These images can be used to identify areas of stress or disease in crops, optimize irrigation and fertilization, and monitor plant growth over time. The camera is often used in conjunction with drones or other aerial vehicles to capture images of large areas quickly and efficiently and this is also its use within IoT-NGIN use case 4.

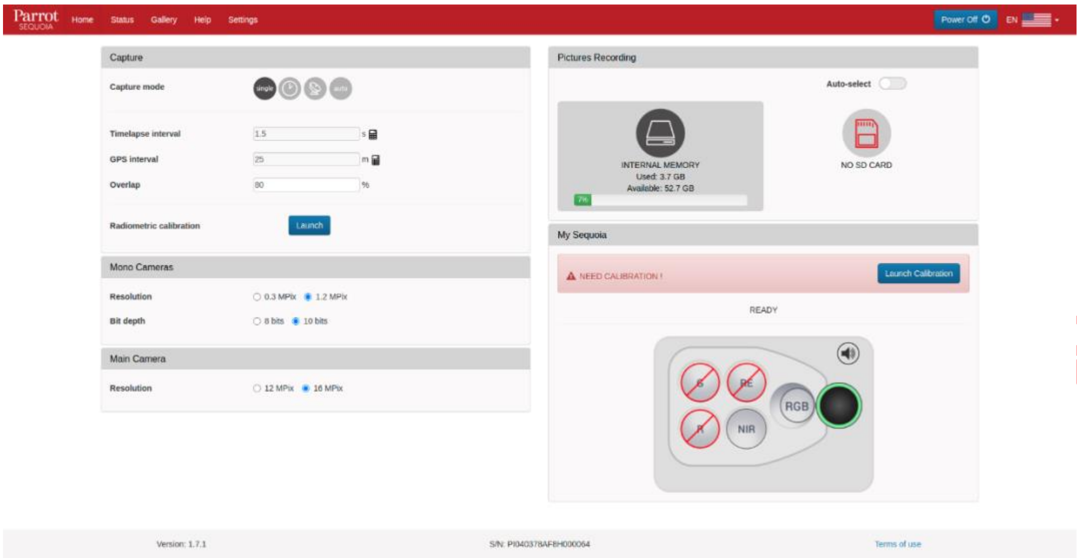


Figure 25: Home screen to set up Sequoia.

The Sequoia camera can be connected to the Jetson board via a USB connection. Once the connection is established, the Sequoia camera's purple light indicates that an initial calibration is required. Calibration is an important process that ensures that the camera is capturing accurate and reliable images. The calibration can be done through a user-friendly interface as shown in Figure 25. The interface provides a range of options, including selecting which spectral bands to include or exclude in the image capture process, such as green, red, and RGB images. Since the model is trained on RGB images, we have configured the Sequoia camera to exclusively capture only RGB images. After the calibration process is complete, the purple light on the camera will turn green, indicating that the camera is ready to capture images. Figure 26 shows the connection of Jetson-Nano with the Sequoia.

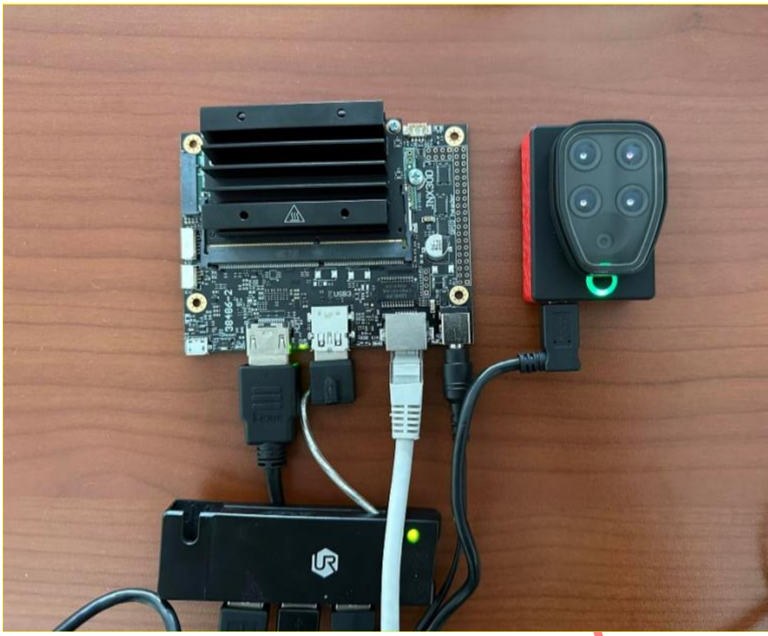


Figure 26: Connection of Sequoia with Jetson-Nano.

3.2.1.8.1.5 Inference process

After successfully establishing the connection between the Sequoia camera and the Jetson Nano, and configuring the necessary settings for the Sequoia, the system is ready to capture images. The primary objective of this setup is to capture high-quality images via the Sequoia camera and process them through the inference service to accurately classify the grapevine leaves as Esca-infected, healthy, or unknown.

The camera operation is controlled through a dedicated script running on the Jetson board, which controls the image captures and ensures that they are available for the inference service. Upon emergence of a new image, the inference service analyses it and provides a classification output based on the characteristics of the image. Figure 27 shows the operation of the inference service.

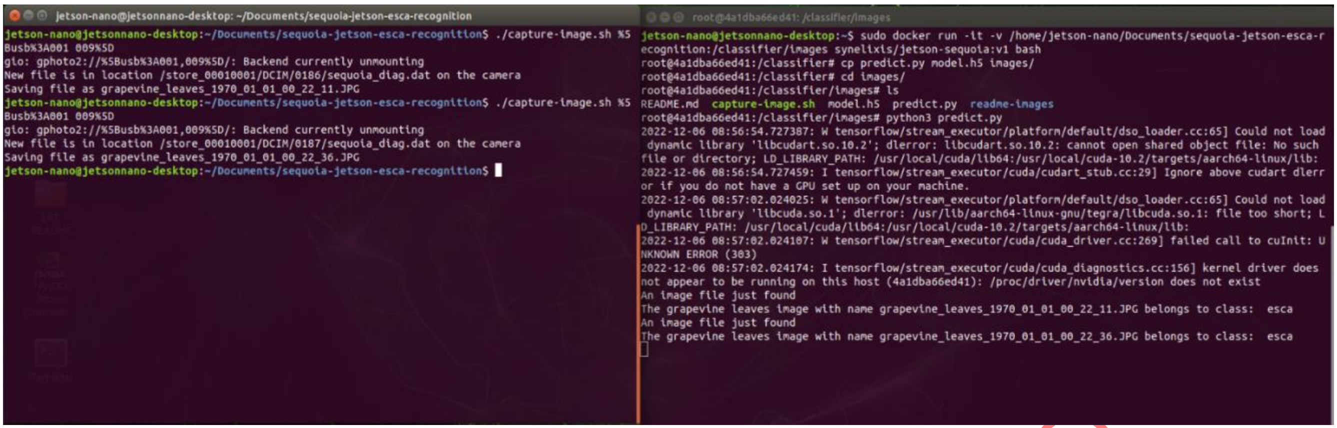


Figure 27: Inference Process; The left side of the screenshot displays the terminal executing the image capture script, while the right side showcases the active inference script.

3.2.1.8.2 Smart agriculture cybersecurity

Within the cybersecurity related tests, vulnerability scanning of the devices connected in the network of the LL and which are within the IC4 interest, such as SynField nodes and UAV, will be performed through the IoT-NGIN Vulnerability Crawler. A deployed instance is already available and operational in this LL K8S cluster, as depicted in Figure 28.

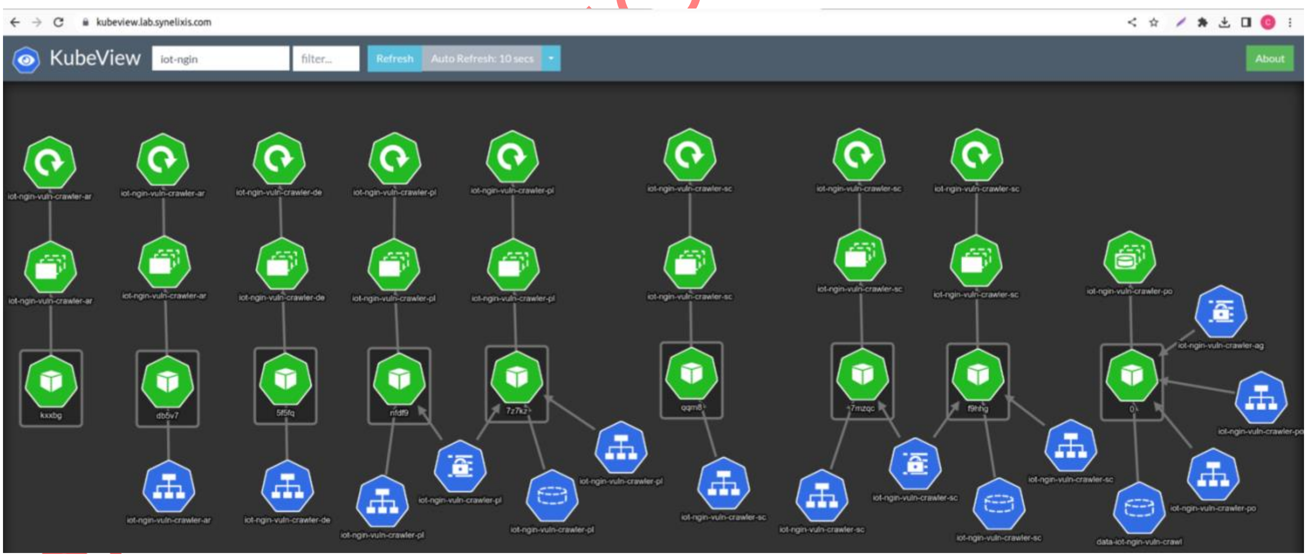


Figure 28: IoT Vulnerability Crawler deployment in the Smart Agriculture LL K8S cluster.

For the detection of malicious activity, IoT-NGIN MAD is employed to monitor network traffic and detect anomalies which are identified as potential attacks. For this purpose, network traffic has to be monitored in the Smart Agriculture LL network. Suricata has been selected and deployed as a tool for monitoring network traffic in this LL.

Suricata [8] is an open source-based network analysis, intrusion detection system (IDS) and intrusion prevention system (IPS) developed by the Open Information Security Foundation (OISF). It implements a complete signature language to match on known threats, policy violations and malicious behavior. Suricata is capable of using the specialized Emerging Threats Suricata ruleset and the VRT ruleset, so it detects many anomalies in the traffic it inspects. Suricata is also used as a Network Security Monitoring (NSM) engine, as it can log HTTP requests and TLS certificates, extract files from flows and store them to disk.

Suricata can be used as an IDS / IPS in a Kubernetes cluster, in order to monitor and protect the cluster network from potential security threats. It can analyze all incoming and outgoing traffic of the Kubernetes cluster, including traffic between the nodes, and external networks. Suricata is configured to capture network traffic and analyze it using a set of predefined rules and custom rules that are created based on the specific security requirements.

The docker image used for the Suricata deployment is the jasonish/suricata [9] which is developed and updated by an OISF member, Jason Ish. The latest stable release version is used, which is the 6.0.10.

To deploy Suricata in the Kubernetes cluster of the Smart Agriculture LL, a DaemonSet has been used in order to ensure that all nodes run a copy of the pod. The securityContext of the container has been set to privileged to gain access to the node network. This allows Suricata to analyze all incoming and outgoing traffic of the Kubernetes nodes' main network interface. The Suricata logs are stored in /var/log/suricata/ directory which is the mount path of a PersistentVolume that has been mounted by all nodes with ReadWriteMany access mode.

Suricata on Kubernetes has been integrated with the Malicious Attack Detector (MAD) component, in order to detect and alert for potential security threats, such as malware, network attacks, and suspicious traffic patterns. This integration is achieved by running MAD as an extra container in the DaemonSet described above, that reads in real time the Suricata output/logs and alerts if a potential security issue is detected.

Figure 29 depicts the Suricata and MAD components deployed in the Smart Agriculture LL K8S cluster.

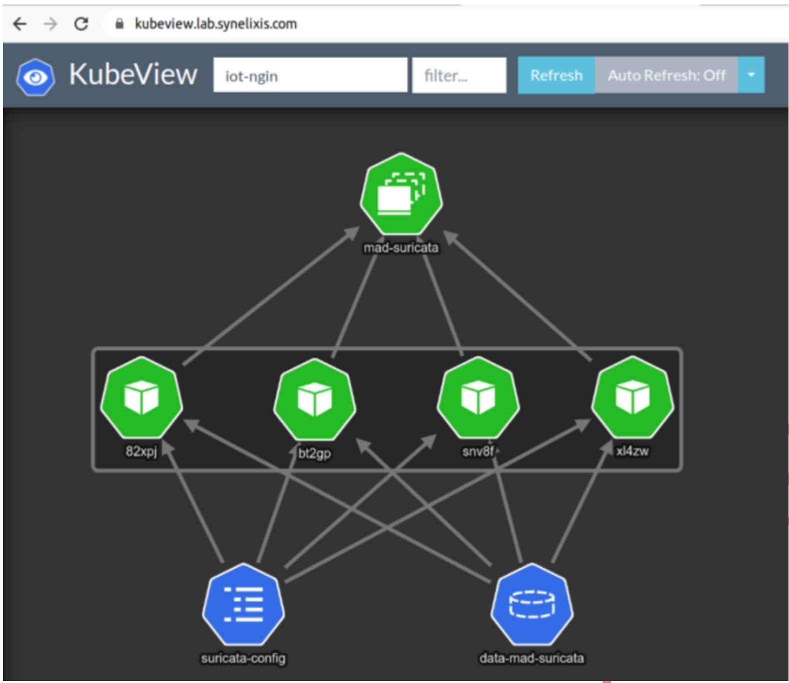


Figure 29: Suricata and Malicious Attack Detector instances in the Smart Agriculture LL K8s cluster.

MAD continuously analyses the network data of Suricata in a streaming fashion and provides the results of this analysis in the form depicted in Figure 30. No malicious activity has been identified so far, so the output is always for benign operation. In the opposite case, the output would notify about "malicious network activity".

```
--- 0.021471261978149414 seconds ---
severity:  3
Benign Network Log
-----------
--- 0.021178483963012695 seconds ---
severity:  3
Benign Network Log
-----------
--- 0.0212404727935791 seconds ---
severity:  3
Benign Network Log
-----------
--- 0.02442455291748047 seconds ---
severity:  2
Benign Network Log
-----------
--- 0.017096996307373047 seconds ---
severity:  3
Benign Network Log
-----------
--- 0.028836727142333984 seconds ---
severity:  3
Benign Network Log
-----------
--- 0.018342971801757812 seconds ---
severity:  3
Benign Network Log
```

Figure 30: Network analysis outcome of MAD in the Smart Agriculture LL network.

3.2.2UC5 – Sensor aided crop harvesting

IoT-NGIN Use Case #5 "**Sensor aided crop harvesting**" aims to facilitate the crop harvesting process, by alleviating human workers from the heavyweight activity of carrying the crates with the harvested crops to the loading points.

In order to achieve this, we suggest an autonomous mobile robot or Automated Guided Land Vehicle (AGLV), able to move safely between the crates' location and the loading point. The main drivers for the realization of this vision include:

- A Smart Agri AGLV implementation, from a hardware perspective, which can be programmatically controlled and be provided as open hardware to the extent possible
- Autonomous operation, which ensures safe movement of the AGLV between the desired locations. The AGLV should be able to reach a given destination, but ensuring that it will always avoid humans, trees, or other obstacles.

In this UC, we experiment with agricultural AGLV serving as carrier machines, which is fully developed in the scope of IoT-NGIN, both on hardware and software perspective. Although there are similar hardware solutions available, they are either lacking the level of programmability or the characteristics for outdoor operation or are proprietary. As a result, we decided to develop an AGLV, fully integrated with IoT-NGIN tools as a far-edge node of the IoT-NGIN ecosystem, which addresses the needs for assisting crop harvesting.

In the next subsections, we provide detailed information on the organization and implementation of the LL validation activities.

3.2.2.1 Trial site description

The use case will be piloted in a commercial vineyard in Peloponnese, Greece, as UC4. The infrastructure elements of UC5 and their interconnections have been defined in D7.2 and are also depicted in Figure 31.

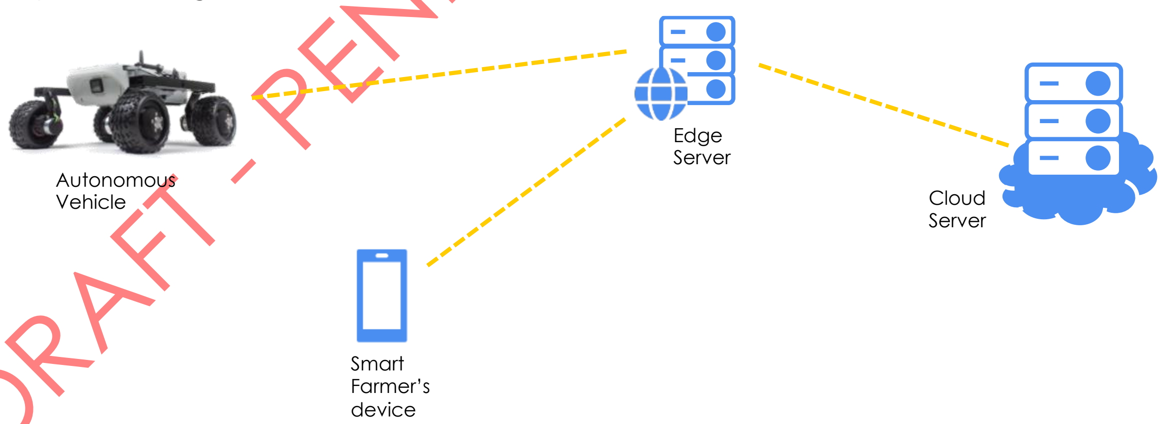


Figure 31: Infrastructure topology for UC5.

3.2.2.2 Required Equipment

3.2.2.2.1 Hardware Description

Table 18 lists the hardware components used in UC5, followed by a brief description of them, along with their type and availability status.

Table 18: Hardware components of UC5.

Equipment	Description / Specifications	Type	Status
Robot	Wild Thumper 6-Wheel platform mounted with additional units such as microcontroller, process units and stereo camera for controlling robot's behavior.	AGLV	Available (1st prototype ready)
Edge node	Edge server for running Digital Twin functionality. The edge node will host the client-side services of the IoT-NGIN framework.	Edge server	Available
Cloud node	Cloud resources used for permanent storage and ML model aggregation/training activities. Also, here the server-side services of IoT-NGIN will be deployed.	Cloud	Available within the Consortium

Table 19 presents a brief description of each component used in the mobile robot or the Automated Guided Land Vehicle (AGLV). The motors of the AGLV have embedded a speed measuring module that is necessary for the autonomous navigation.

Table 19: Detailed description of the AGLV components.

Equipment	Description / Specifications	Type	Status
Robot chassis	Wild Thumper 6-Wheel platform. Each of the six motors are mounted on independent suspension and outfitted with 34:1 steel gearbox. The entire chassis is perforated with 10mm pitched, 4mm diameter mounting holes and there is plenty of room interior to the chassis for batteries, drivers and other support hardware.	AGLV	Available
SynBoard	The SynBoard is a microcontroller board with 8 sensors' connectors, 8 actuators' connectors a SIM card socket, 4 DIP	Microcontroller	Available

	switches, a battery connector and an ON/OFF switch. It is responsible for executing the commands related to the AGLV's main operation, i.e. movement.		
Ultrasonic sensor A02YYUW	A02YYUW ultrasonic distance sensor. providing 3cm to 450cm of non-contact measurement functionality with a ranging accuracy up to 3mm.	Sensor	Available
Processing unit	NVIDIA Jetson Xavier NX 16GB, a powerful computer which is able to run multiple neural networks in parallel for applications like object detection, which is desired for the autonomous navigation.	Processor	Available
Arcmax PPSMAX 600	Rechargeable battery for powering up the NVIDIA Jetson Xavier NX. Arcmax PPSMAX 600 can produce a 12V/3A output which is required for Jetson Xavier's proper functionality.	Battery	Available
TP-Link AC1300 Wi-Fi adapter	A high-speed Wi-Fi adapter that allows the connection of the Jetson Xavier NX to a wireless network. It supports dual-band Wi-Fi connectivity, which means it can connect to both 2.4GHz and 5GHz networks, providing faster and stable connections.	USB Wi-Fi adapter	Available
Zed 2i	IP66 stereo camera, wide-angle 3D AI camera, multiple lens selection with polarizer, built-in IMU, barometer & magnetometer.	Camera	Available

The AGLV parts listed in Table 18 have been assembled, following the AGLV design presented in Figure 32. Autonomous navigation of the AGLV requires it to detect the obstacles in its route. Relevant services running in the processor along with the stereo camera cater for this operation. The NVIDIA Jetson Xavier NX processor has the computational power to run Neural Networks with GPU acceleration and thus perform ML object detection for detecting the obstacles captured via the Zed2i camera. The Zed2i camera supports distance measurement of specific areas of the image with respect to the camera itself. In this way, it is possible to compute the distance (depth) between the detected objects and the AGLV.

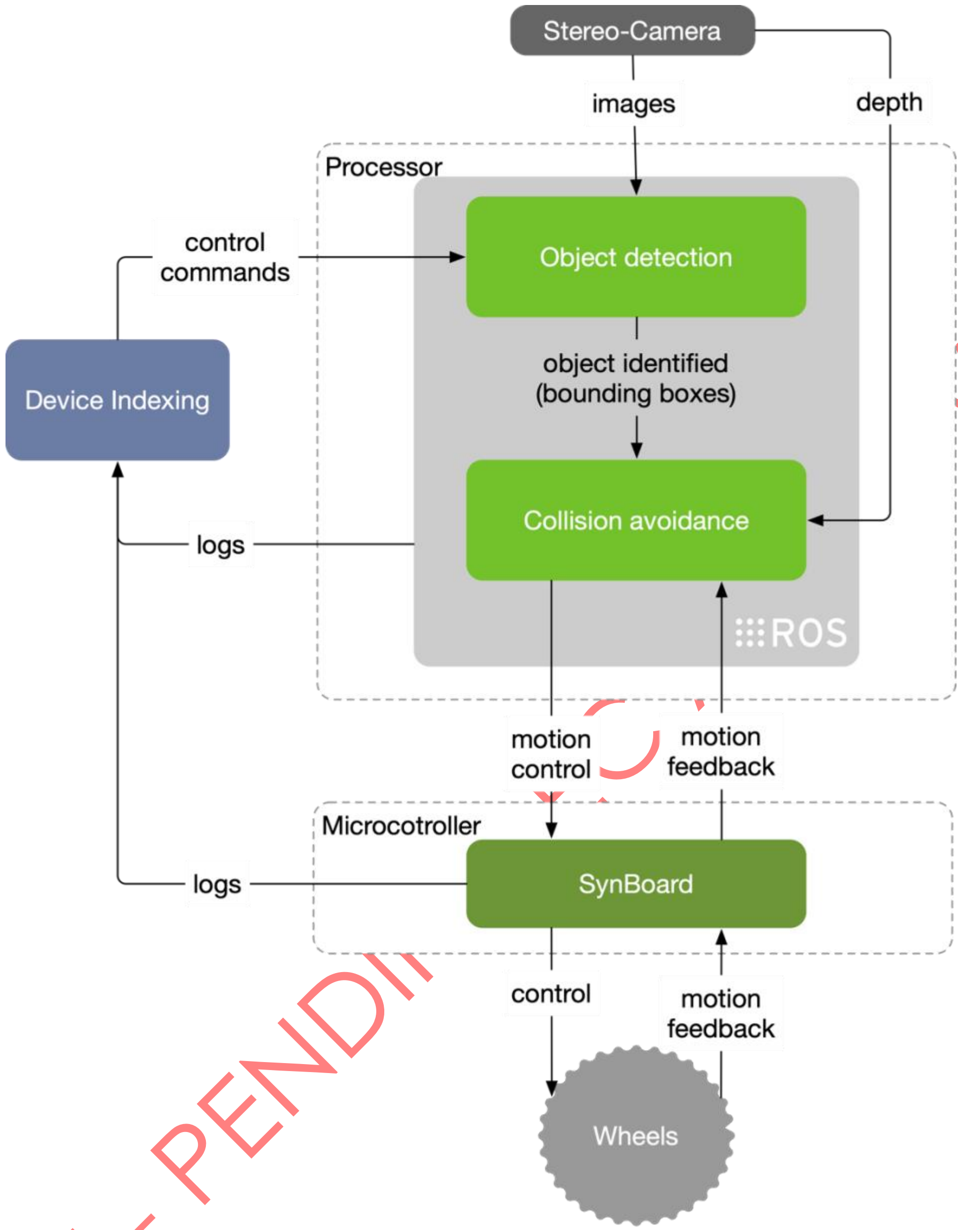


Figure 32: The AGLV component-level design.

Accordingly, the Processor is processing the data from the object detection ML model through a collision avoidance algorithm developed within IoT-NGIN, and then sends commands to the microcontroller (SynBoard) in order to control the AGLV movement. These commands refer to how much each wheel of the AGLV should move and are in range (-255, 255), where negative values correspond to backward movement and positive values to forward movement of the AGLV. The SynBoard then, is responsible for transmitting these commands to the motors of the AGLV and thus the AGLV is moving. More details about the software are presented in the next section.

The assembled AGLV prototype is shown in Figure 33 (side image) and Figure 34 (front image).

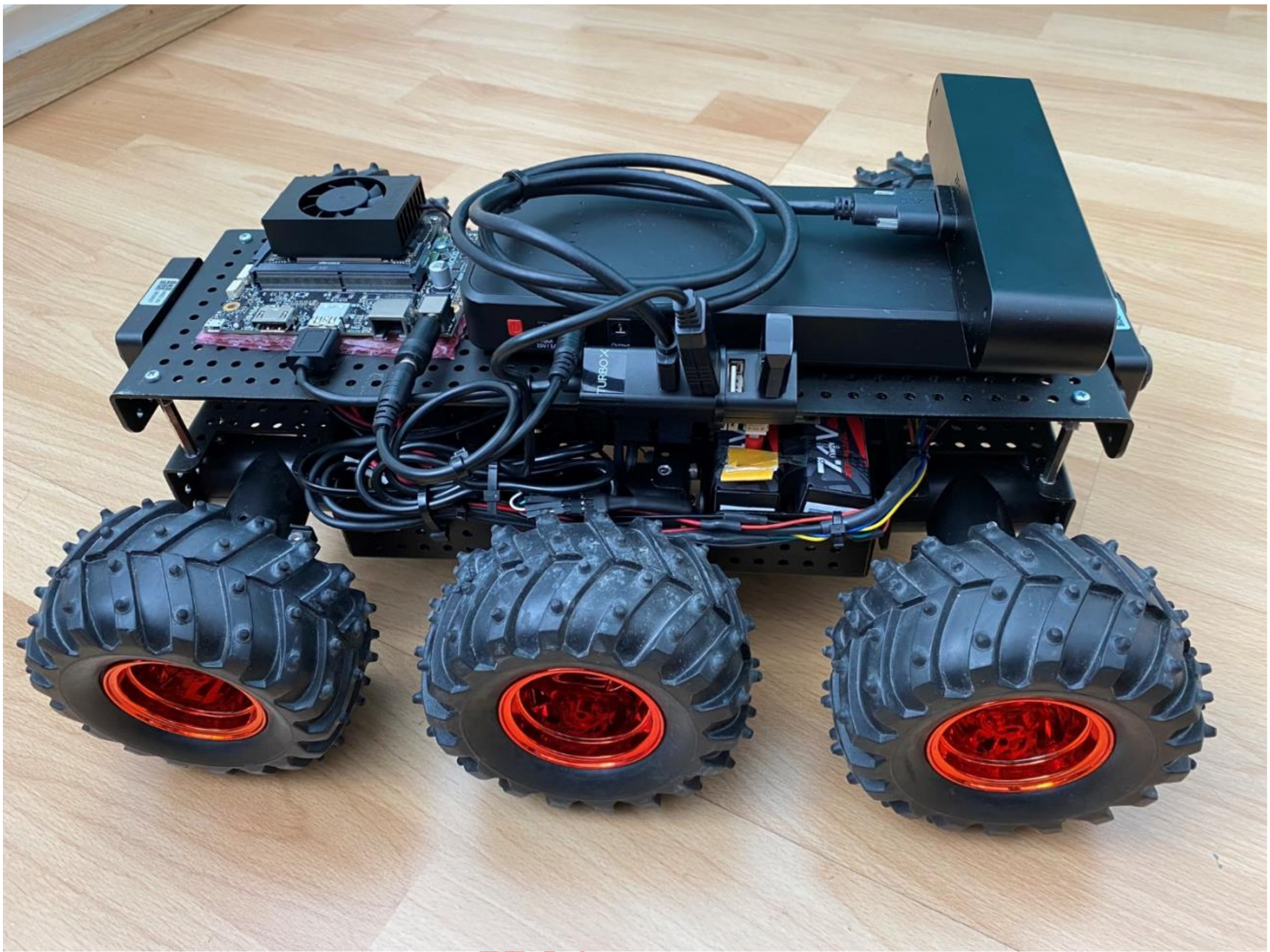


Figure 33: Side image of the AGLV prototype developed within IoT-NGIN.

Figure 34: Front image of the AGLV prototype developed within IoT-NGIN.

The Jetson Xavier NX has been connected with other parts, including the Zed2i stereo camera, the TP-Link AC1300 Wi-Fi adapter and the SynBoard. The Wi-Fi adapter is used for internet access of the Jetson Xavier NX processor while it also offers connection to the local network facilitating development on the Jetson. In addition, the SynBoard is also connected to the Jetson board through a USB-to-TTL Serial Cable. TTL stands for Transistor-Transistor Logic, which is a type of digital circuit that uses transistors to implement logic gates. The communication protocol used in a USB-to-TTL serial cable is typically the UART (Universal Asynchronous Receiver/Transmitter) protocol. UART is a popular protocol for asynchronous serial communication between devices. It enables data to be transmitted and received on two data lines, one for transmitting data (TX) and one for receiving data (RX), using a common ground reference. In this way, data can be successfully sent from the Jetson Xavier NX processor to the SynBoard microcontroller.

SynBoard is comprised of a single printed circuit board (PCB). SynBoard is a microcontroller board based on ATmega 1264p. Figure 35 shows the layout of the device's PCB along with its connectors and switches. The board includes the following connectors and switches:

- **ON/OFF switch** for powering ON and OFF the device
- **Solar-Panel Connector** for plugging in either the solar or an external 5VDC power-supply
- **Battery connector** for connecting the device battery
- **Sensors' connectors**:
  - connectors 1-4 that support analog (10-bit accuracy), pulse counters and unidirectional serial sensors
  - connectors 5-6 that support analog sensors (16-bit accuracy)
  - connector 7m
  - connector 8d for digital sensors (UART, I2C, SDI12)
- **Actuators' connectors** A1-A8 for installing valves and bi-stable relays (with 1 or 2 coils)
- **SIM Card socket** for inserting cellular provider SIM.
- **DIP switch 1**: Configuring connectors 1 to 4 as analog or pulse-counter sensors' ports
- **DIP switch 2:** Configuring the output voltage-level of the digital sensor (connector 8d) to 3.3V or 5V
- **DIP switch 3:** Configuring the output voltage-level of the actuators to 9V or 12V
- **DIP switch 4:** Configuring connectors 5 and 6
- **GPS antenna connector** for installing an active, 3V, GPS antenna (at the back side of the PCB)
- **Cellular antenna connector** for optionally installing an external cellular antenna (typically not used). (at the back side of the PCB)

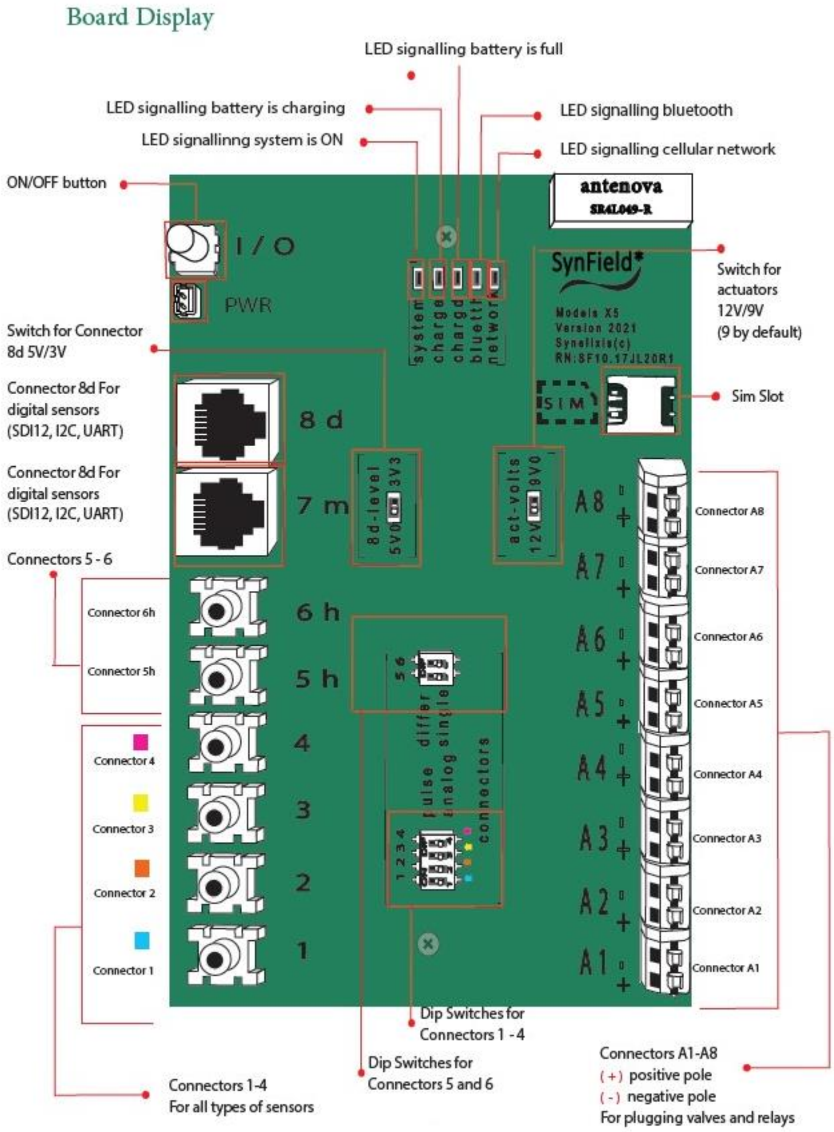


Figure 35: Synboard Printed Circuit Board.

3.2.2.2.2 Software Description

The entire system of the AGLV's navigation process is developed in ROS (Robot Operating System) [10]. ROS is an open-source framework for building robot applications. It provides libraries and tools to help developers create software for robots that can run on a variety of hardware platforms and operating systems. ROS was originally developed by Willow Garage, but it is now maintained by the Open Robotics organization [11]. ROS operates on a distributed system architecture, where multiple nodes (software processes) can communicate with each other through a communication framework. The communication framework enables nodes to send and receive messages, and to subscribe to and publish data streams. A more detailed architecture of ROS is presented in Figure 36. Nodes in ROS can be written in various programming languages such Python and C++. They can run on different hardware platforms and communicate with each other through standard communication protocols. This makes it easy to develop software for robots that can operate across different platforms. ROS also provides a variety of tools and libraries to simplify the development of robot applications as well as visualization tools to help developers debug and test their robot applications.

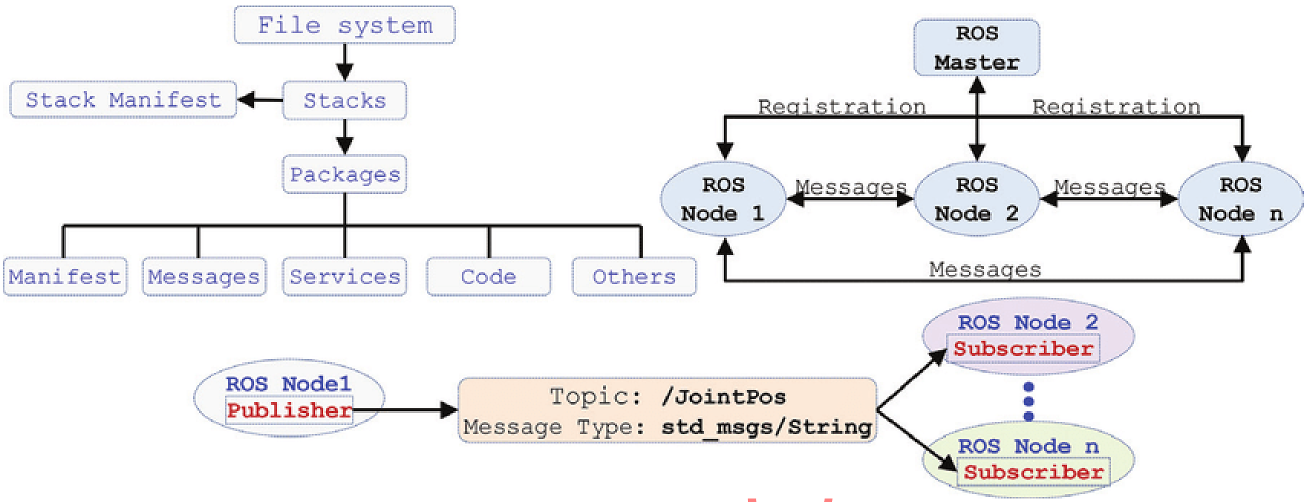


Figure 36: Diagram showing ROS file architecture and nodes communicating system, Source: [12].

The publish-subscribe architecture is a fundamental concept in ROS that is used for communication between nodes. This architecture enables nodes to communicate with each other in a decentralized and asynchronous manner, which is well suited for robotic applications.

The AGLV services within both Jetson Xavier NX and SynBoard execute everything within ROS. Initially, the stereo camera Zed2i is being activated through ZED-ROS-Wrapper [13], a package developed from Stereolabs in order to exploit all the capabilities of the Zed2i camera within the ROS framework. After launching the ZED-ROS-Wrapper package, multiple topics are generated, one topic for each information that the camera can provide such as stereo images, depth maps, odometry data etc. Figure 37 shows a portion of the generated topics when launching the ZED-ROS-Wrapper package within ROS in the AGLV's Jetson Xavier processor. To ensure that the camera is working properly, we can use the ROS Visualization (RVIZ) tool [14].

```
[ INFO] [1678805715.671042094]:  * CAMERA MODEL  -> ZED 2i
[ INFO] [1678805715.671334711]:  * Serial Number -> 37417973
[ INFO] [1678805715.671550430]:  * Camera FW Version -> 1523
[ INFO] [1678805715.672454651]:  * Sensors FW Version -> 777
[ INFO] [1678805716.442674587]: Advertised on topic /zed2i/zed_node/rgb/image_rect_color
[ INFO] [1678805716.442846144]: Advertised on topic /zed2i/zed_node/rgb/camera_info
[ INFO] [1678805716.528385845]: Advertised on topic /zed2i/zed_node/rgb_raw/image_raw_color
[ INFO] [1678805716.528537082]: Advertised on topic /zed2i/zed_node/rgb_raw/camera_info
[ INFO] [1678805716.612679045]: Advertised on topic /zed2i/zed_node/left/image_rect_color
[ INFO] [1678805716.612895724]: Advertised on topic /zed2i/zed_node/left/camera_info
[ INFO] [1678805716.696507719]: Advertised on topic /zed2i/zed_node/left_raw/image_raw_color
[ INFO] [1678805716.696690860]: Advertised on topic /zed2i/zed_node/left_raw/camera_info
[ INFO] [1678805716.780359369]: Advertised on topic /zed2i/zed_node/right/image_rect_color
[ INFO] [1678805716.780595440]: Advertised on topic /zed2i/zed_node/right/camera_info
[ INFO] [1678805716.860201970]: Advertised on topic /zed2i/zed_node/right_raw/image_raw_color
[ INFO] [1678805716.860378264]: Advertised on topic /zed2i/zed_node/right_raw/camera_info
[ INFO] [1678805716.937797208]: Advertised on topic /zed2i/zed_node/rgb/image_rect_gray
[ INFO] [1678805716.938043359]: Advertised on topic /zed2i/zed_node/rgb/camera_info
[ INFO] [1678805717.021500560]: Advertised on topic /zed2i/zed_node/rgb_raw/image_raw_gray
[ INFO] [1678805717.021818969]: Advertised on topic /zed2i/zed_node/rgb_raw/camera_info
[ INFO] [1678805717.097275055]: Advertised on topic /zed2i/zed_node/left/image_rect_gray
[ INFO] [1678805717.098289356]: Advertised on topic /zed2i/zed_node/left/camera_info
[ INFO] [1678805717.173881223]: Advertised on topic /zed2i/zed_node/left_raw/image_raw_gray
[ INFO] [1678805717.174324915]: Advertised on topic /zed2i/zed_node/left_raw/camera_info
[ INFO] [1678805717.246903384]: Advertised on topic /zed2i/zed_node/right/image_rect_gray
[ INFO] [1678805717.247364357]: Advertised on topic /zed2i/zed_node/right/camera_info
[ INFO] [1678805717.322688664]: Advertised on topic /zed2i/zed_node/right_raw/image_raw_gray
[ INFO] [1678805717.323728982]: Advertised on topic /zed2i/zed_node/right_raw/camera_info
[ INFO] [1678805717.395791212]: Advertised on topic /zed2i/zed_node/depth/depth_registered
[ INFO] [1678805717.396017299]: Advertised on topic /zed2i/zed_node/depth/camera_info
[ INFO] [1678805717.466177876]: Advertised on topic /zed2i/zed_node/stereo/image_rect_color
[ INFO] [1678805717.535314392]: Advertised on topic /zed2i/zed_node/stereo_raw/image_raw_color
[ INFO] [1678805717.537938434]: Advertised on topic /zed2i/zed_node/confidence/confidence_map
[ INFO] [1678805717.540802259]: Advertised on topic /zed2i/zed_node/disparity/disparity_image
```

Figure 37: Generated topics from Zed2i stereo camera within ROS for the UC5 AGLV.

RVIZ is a powerful 3D visualization tool in ROS (Robot Operating System) that is used to display and interact with sensor data, robot models, and other data in a 3D environment. RVIZ is designed to help developers and researchers visualize and debug complex robot systems. and allows users to view sensor data from various sources, such as cameras, lidars, and depth sensors. It can display data in various formats, such as point clouds, images, and laser scans. and also provides tools to manipulate and filter sensor data, making it easy to analyze and understand complex data sets. Figure 38 shows an example of using the RVIZ visualization tool to display the images from the Zed2i camera. In a high-level overview, RVIZ subscribes to the topic */zed2i/zed_node/left/image_rect_color* and displays the image.

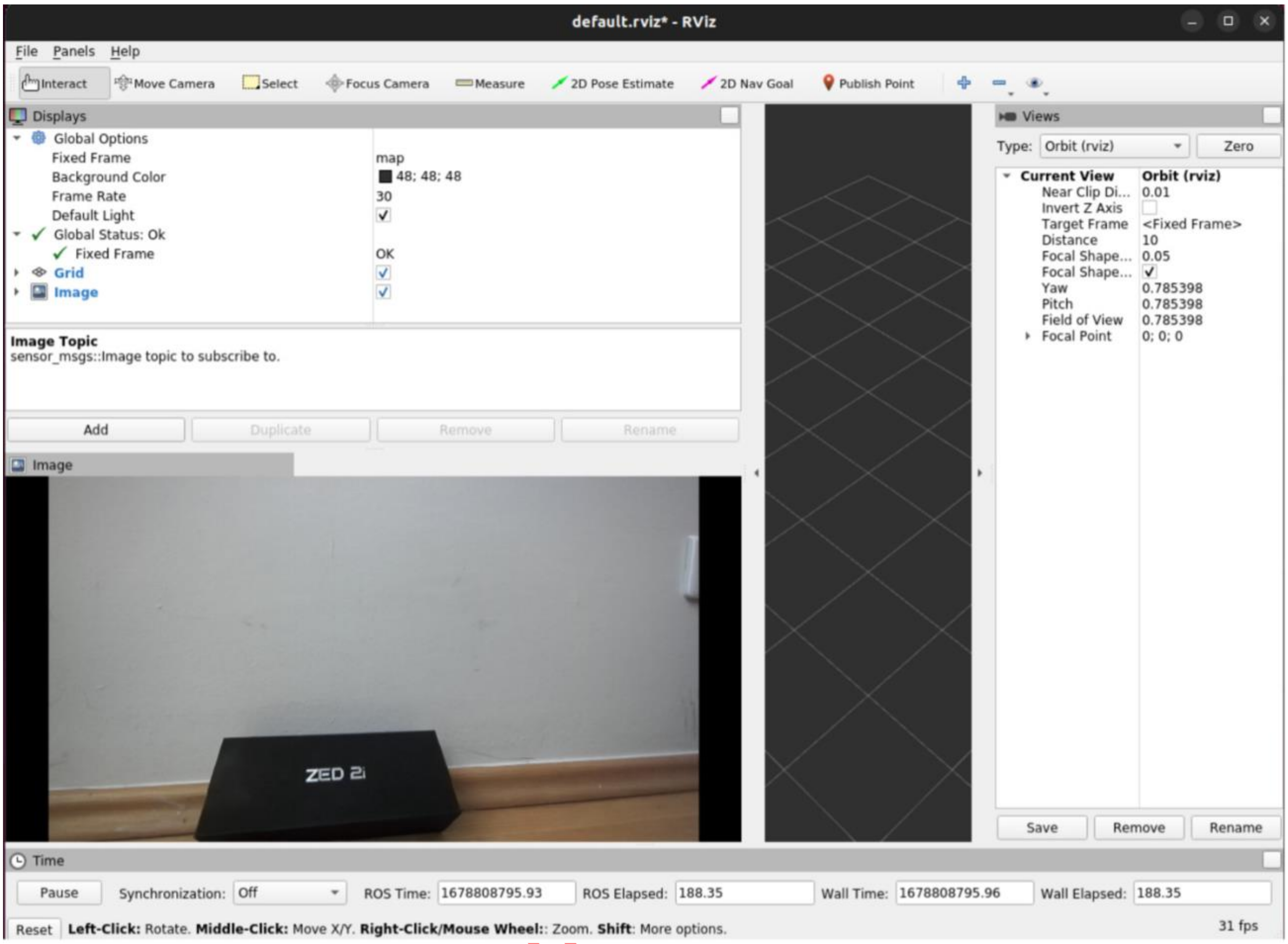


Figure 38: Example of displaying Zed2i camera images using the RVIZ visualization tool within ROS.

Afterwards, the Jetson Xavier NX processor takes as input the images from Zed2i through the ROS framework, runs object detection for detecting the obstacles and generates the navigation commands for the AGLV's motors using our collision avoidance algorithm. All the tests for the AGLV's autonomous navigation were performed using the Gazebo simulation [15].

Gazebo is a widely used open-source 3D robotics simulator that is integrated with ROS. It allows simulating the motion and behavior of multiple robots and sensors in a virtual environment. Gazebo simulates the physics of robots and their interactions with the environment, allowing developers to test their algorithms, control systems, and sensors without the need for expensive physical prototypes. It supports a wide range of robots, sensors, and other hardware components, making it an ideal platform for developing and testing complex robotic systems. Figure 39 shows an example of a custom Gazebo environment. The white robot at its centre simulates our AGLV and is needed in order to perform various tests and develop and test the autonomous navigation algorithm.

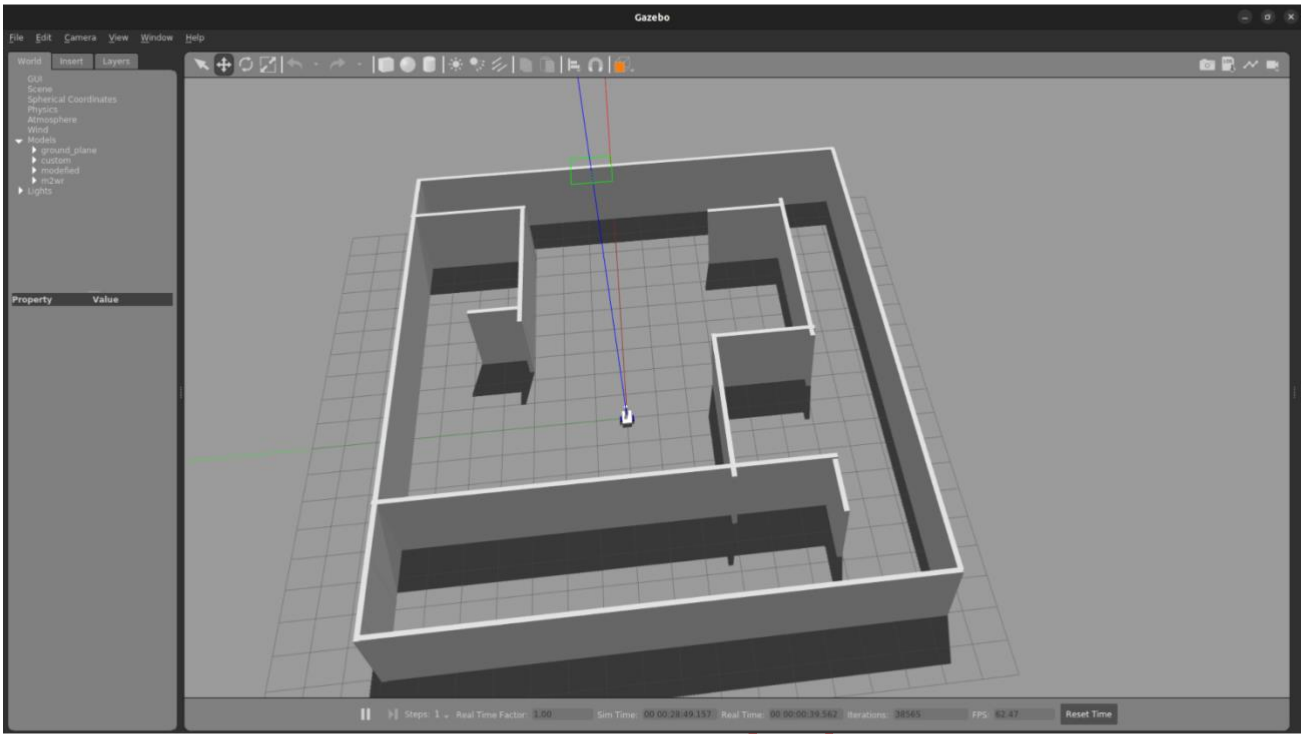


Figure 39: Gazebo environment with a simulated robot at the centre of the environment.

3.2.2.2.3 Obstacle Detection

The AGLV can successfully detect the obstacles that may occur during its navigation procedure using the ML Yolov5 [16] object detection algorithm. We finetune Yolov5 in a federated manner for detecting only persons as in UC5 the AGLV is assumed to potentially face only humans during its navigation. For the finetuning procedure we use the Pascal Voc (Visual Object Classes) [17] dataset, a publicly available dataset for object detection. Yolov5 has already been trained in the COCO dataset [18] and we exploit these trained parameters for finetuning. That said, the final finetuned model is capable of detecting "persons" more accurately. The training process (i.e. the finetuning) is being initiated through the Privacy Preserving Federated Learning API (PPFL-API) developed in WP3 and documented in D3.4, while the training process per se is performed through the NVIDIA FLARE Federated Learning Framework enhanced with privacy preservation, as investigated in D3.3 [19].

For the selection of the appropriate Yolov5 architecture[3] (Yolov5n, Yolov5s, Yolov5m, Yolov5l) we performed various tests in the Gazebo simulation under the following scenarios:

- *Scenario 1*. Maximum distance of detectable object, i.e., how far can the algorithm detect the obstacles.
- *Scenario 2*. Object detection granularity, i.e., evaluating the ability of the algorithm to identify the obstacle when a portion of the obstacle is shown in the image.

We have integrated Yolov5 in ROS in order to test it in the Gazebo simulation and perform obstacle detection in our custom environment. We experiment with the Yolov5s and Yolov5m model architectures under several scenarios. Gazebo does not contain persons and thus we

[3] Yolov5 releases can be found here: https://github.com/ultralytics/yolov5/releases.

could not simulate them. That said, the simulated obstacles are represented as Stop Signs during AGLV's navigation procedure in our Gazebo simulation.

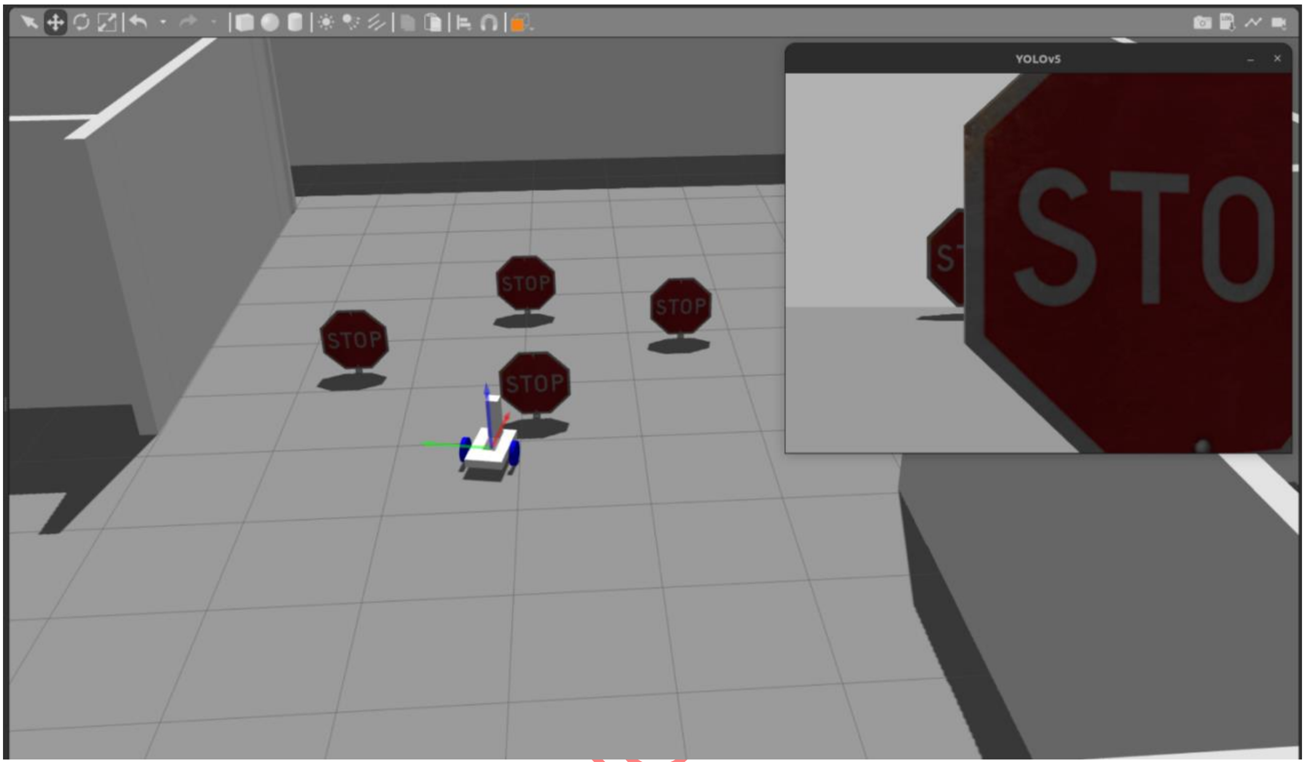


Figure 40: Test Yolov5n architecture under Scenario 1, the distance of the objects.

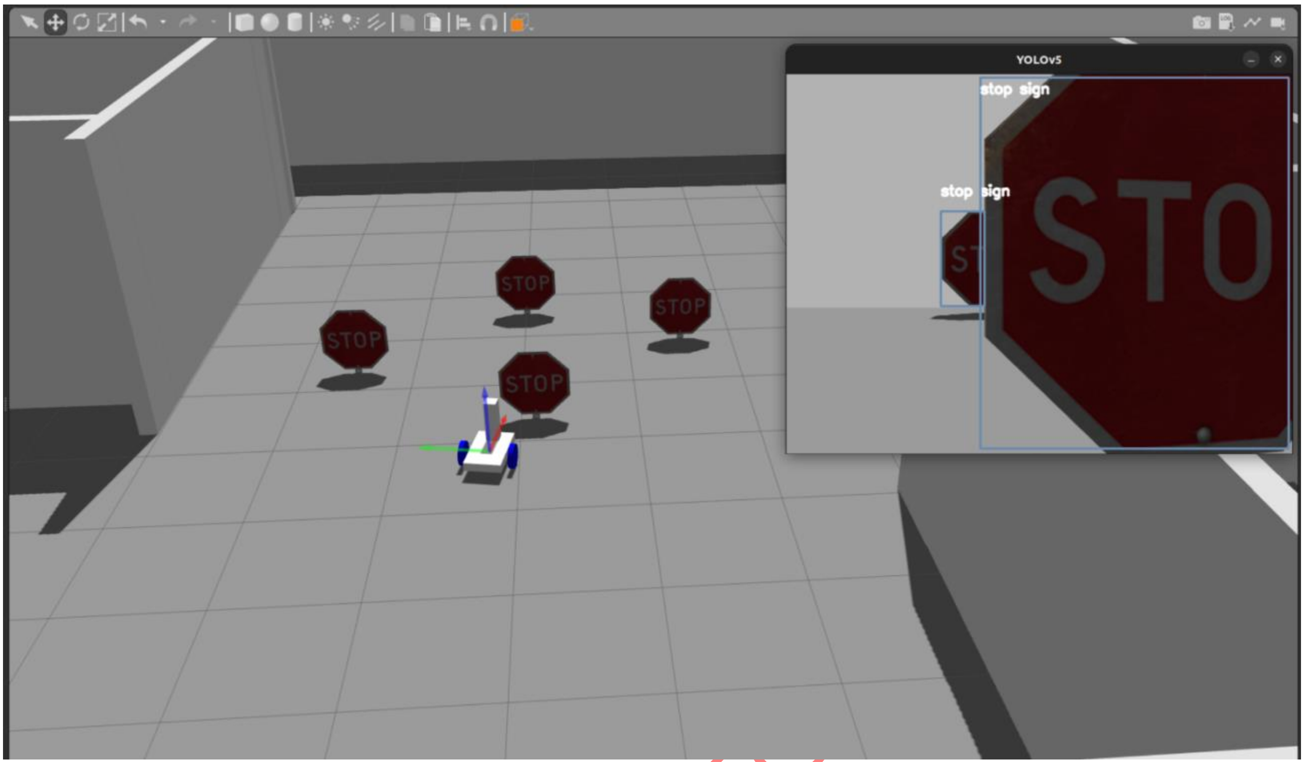


Figure 41: Test Yolov5s architecture under Scenario 1, the distance of the objects.

Figure 40 and Figure 41 show the results of using the Yolov5n and Yolov5s architectures for object detection, respectively, under the Scenario 1. The two-wheeled robot simulates the AGLV and the stop signs the obstacles. The window on the upper right part of the image shows the results of the Yolov5 object detection algorithm while the robot is moving. We observe that the Yolov5n cannot detect the "stop sign" obstacle, neither the distant nor the closest one. On the other hand, the Yolov5s architecture can detect both obstacles successfully. For Scenario 2, we experiment with the Yolov5s and Yolov5m architectures as Yolov5n cannot meet our requirements of detecting obstacles at a short or long distance. Both Figure 42 and Figure 43 show the result of the Yolov5 object detection algorithm when we force the camera of the robot to capture images that contain a small portion of the obstacle. We observe that Yolov5s cannot recognize the obstacle when it barely sees it, while Yolov5m can successfully detect it. That said, Yolov5m meets our requirements, and we will finetune this model for detecting humans.

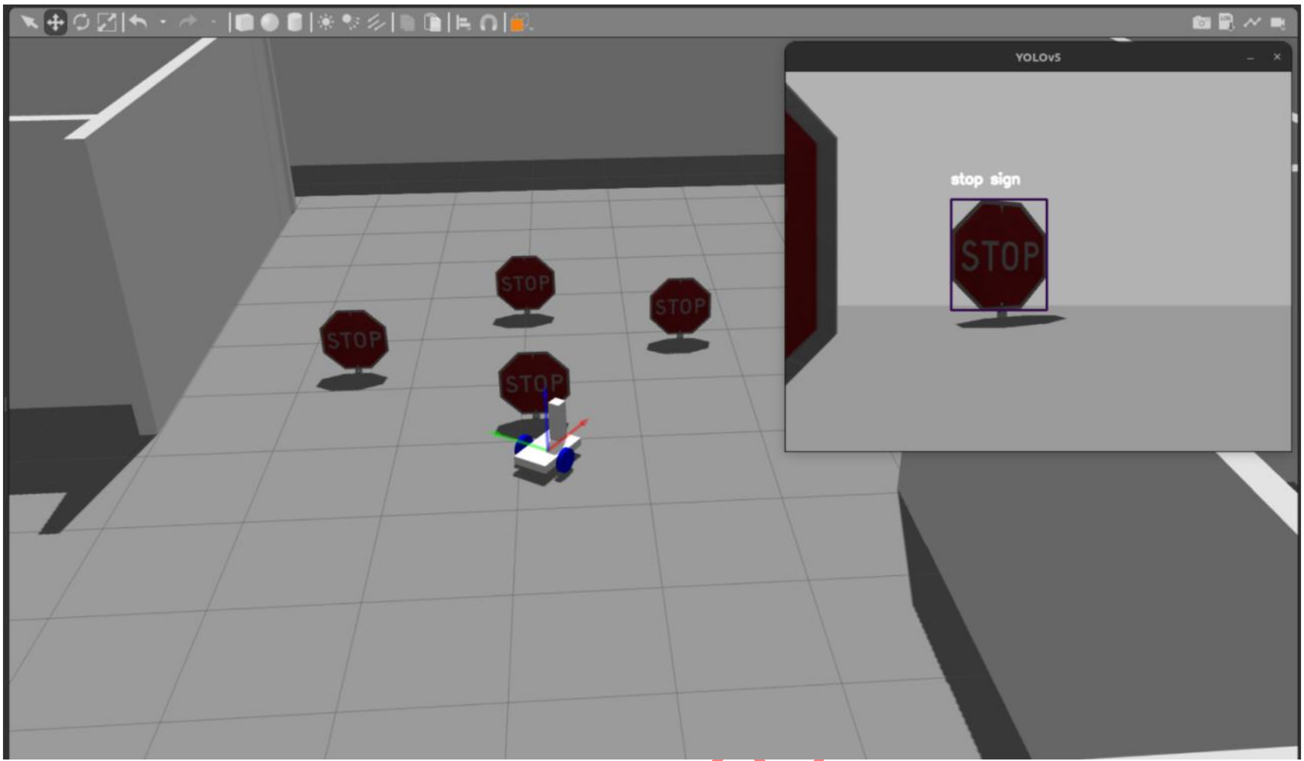


Figure 42: Test Yolov5s architecture under Scenario 2, the performance of Yolov5 when a small portion of an image is displayed.

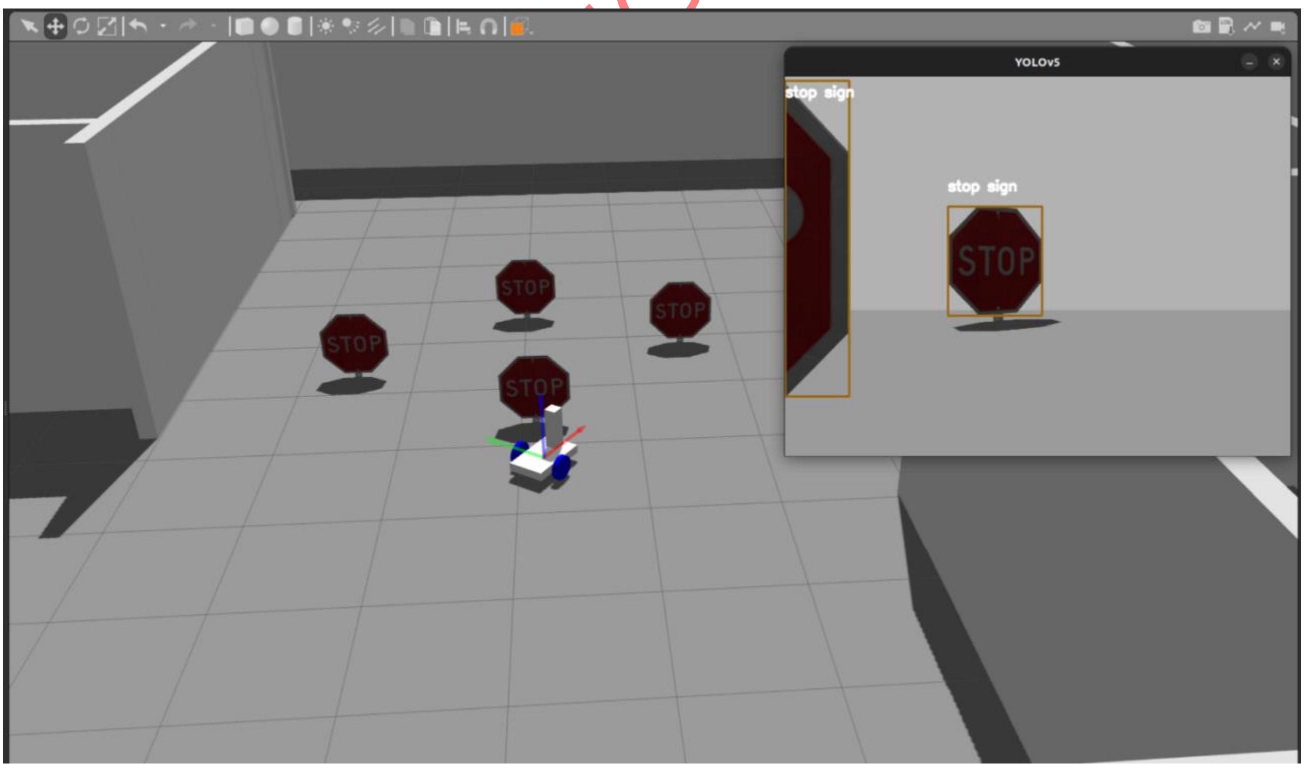


Figure 43: Test Yolov5m architecture under Scenario 2. Performance of Yolov5 when a small portion of an image is displayed.

In the next subsection, the collision avoidance algorithm is described that uses the Yolov5m architecture for obstacle detection.

3.2.2.2.4 Collision Avoidance Algorithm

The AGLV's autonomous navigation is performed by a collision avoidance service that uses the obstacle avoidance algorithm and generates the respective commands for the AGLV movement. Particularly, the collision avoidance service incorporates two main functionalities, namely the navigation without obstacles as well as the obstacle avoidance procedure.

Regarding the navigation part, the AGLV uses Odometry data in order to be aware of its position and orientation in space. The Gazebo simulation provides the Odometry data about the AGLV through the */odom* topic making it easy for the developers to subscribe from it and exploit the data. However, in our case the Zed2i stereo camera provides us this kind of information through its integrated sensors, including stereo cameras, an inertial measurement unit (IMU), and a barometer. The IMU on the Zed2i camera provides inertial odometry data by measuring the camera's linear and angular acceleration and rotation rates and the barometer provides altitude measurements, which can be used in conjunction with the odometry data to provide a more accurate estimate of the camera's position and motion. In ROS, the odometry data from the camera are provided from the */zed2i/zed_node/odom* topic, which is generated when the ZED-ROS-Wrapper package is launched. Subscribing to the odometry data from the corresponding ROS topic, we are able to compute the current orientation and position of the AGLV in the space using the *transformations* library. That said, knowing the coordinates of the goal point we compute the yaw that the AGLV must have in order to move to the desired direction. When the desired direction is known and the AGLV has turned to this direction, it starts to move straight ahead, while we compute the distance of the AGLV from the goal point in order to stop the AGLV if the distance gets too close to 0.

Regarding the avoidance of the detected obstacle per se, we divide the image with the detected obstacle into 3 columns and then the algorithm focuses on 5 points, the left, the middle left, the centre, the middle right and the right point of the bounding box as Figure 44 shows. Afterwards, we compute the distance of the camera with each of the 5 points and we keep the ones that their distance is less than a threshold value (e.g. 0.8 m). Then we calculate in which part of the image (left, middle or right) these points are, and the algorithm decides where the AGLV must turn in order to avoid it. For example, in Figure 44 the closest points are at the middle and the right part of the image so the AGLV must turn left in order to avoid the obstacle. The AGLV will keep turning to the desired direction until there are no points whose distance are below the threshold (i.e. there are no obstacles). When the turning procedure is done, the AGLV will move forward for a short period of time and then it will use the first part of the algorithm, unless it detects another obstacle. The distance of each point from the camera is calculated through a depth map that Zed2i stereo camera provides. This depth map can be found in the */zed2i/zed_node/depth/depth_registered* topic. The depth map is a map that has the same dimensions as the image and contains the distance of each pixel of the image with respect to the camera. So, if we are aware of the coordinates of a point in the image, we can easily compute its distance by taking the value of the corresponding pixel in the depth map.

Overall, while no obstacles are detected, the AGLV movement relies on the navigation part. On the contrary, if an obstacle is detected, then the AGLV movement is guided by the obstacle avoidance part.

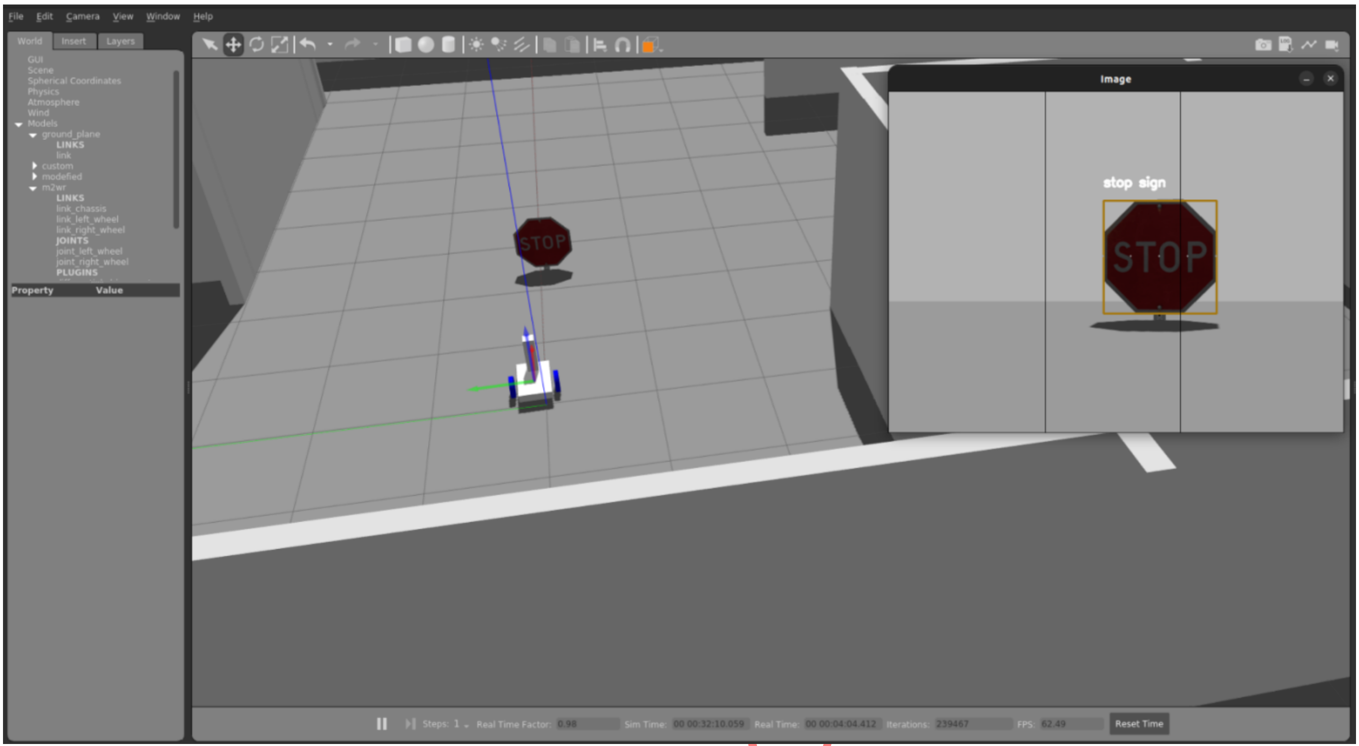


Figure 44: Visualization of the way the robot processes the detected obstacle.

3.2.2.3 Data collection

The following dataset(s) have been identified since D7.2 as part of the use case:

- **AGLV sensor measurements:** measurements acquired from sensors installed and configured specifically for the trial on Automated Guided Land Vehicles (AGLVs) will be collected. The data will be used to allow the use of the AGLVs as carrier machines, and to enable them to locate and avoid workers and trees.
- **AGLV camera images:** video images collected from AGLV's camera. The images will be used for obstacle avoidance of the AGLV.

3.2.2.4 Alignment with IoT-NGIN technologies

Table 20 below shows the IoT-NGIN technologies relevant for UC5, providing updates compared to D7.2. For each technology, a description of its role is provided along with any adaptation it might need to be used in the UC, deployment details and related UC requirements and KPIs.

Table 20: Alignment of UC5 with the relevant IoT-NGIN technologies.

WP2 – Secure edge-cloud execution framework	
Description	This framework will be used in order to enable secure deployment and operation of IoT-NGIN services. Indicatively, it will host the obstacle detection models deployed on the AGLV.
Adaptation and fine-tuning	The Secure Execution framework will be integrated with the MLaaS platform, enabling secure deployment of ML models. As such, no adaptation is needed.
Deployment	Multi-layer, including edge and cloud resources.
Related requirements	**REQ_SA2_F06** – The platform must ensure collected data sovereignty and integrity **REQ_SA2_NF01** – The IoT-NGIN platform must respect security and privacy requirements
Related KPIs	**KPI_SA2_03** – Reduction of the time needed for carrying crates to the loading point, compared to manual carrying >/= 10%
WP3 – MLaaS framework	
Description	The MLaaS framework will be used to provide ML/AI services to the use case, related to ML model training, model storage and model sharing, in order to support the obstacle avoidance application.
Adaptation and fine-tuning	No adaptation of the MLaaS framework is needed for the purposes of UC5. The obstacle avoidance application will consume the MLaaS services interacting with its API.
Deployment	A project-wide deployment can be used. The use case does not require a dedicated instance of the MLaaS framework.
Related requirements	**REQ_SA2_F01** – Mobile robots must support autonomous operation. **REQ_SA2_F10** – The platform must be able to calculate mobile robots' routes in real-time. **REQ_SA2_F11** – The platform must be able to identify and avoid obstacles such as humans and trees in real-time.
Related KPIs	**KPI_SA2_01** – Collisions of mobile robots with human workers = 0.

	KPI_SA2_02 – System reaction in emergency cases < 1 sec.
WP3 – Privacy-preserving Federated Machine Learning	
Description	This framework will be used in order to allow training the ML models at multiple edge nodes in a federated-distributed manner. The locally trained models will be aggregated at the cloud server. The training refers to object detection models.
Adaptation and fine-tuning	The FL framework will be configured to work with a number of "participants" (clients) equal to the number of edge nodes used in UC4.
Deployment	Multi-layer, including edge and cloud resources, as described above.
Related requirements	**REQ_SA1_NF01** – The IoT-NGIN platform must respect security and privacy requirements.
Related KPIs	**KPI_SA2_01** – Collisions of mobile robots with human workers = 0. **KPI_SA2_02** – System reaction in emergency cases < 1 sec.
WP4 – IoT device indexing	
Description	This module will allow accessing data collected by the AGLV, as well as any information accompanying these devices. Moreover, it can be used to dispatch actuation/configuration commands on the AGLVs, such as the deployment of a new ML model on the AGLV.
Adaptation and fine-tuning	No adaptation need identified at this point.
Deployment	Edge
Related requirements	**REQ_SA2_F02** – Mobile robots should be able to understand which crates they should carry **REQ_SA2_F03** – Mobile robots must be able to follow a route or reach a destination **REQ_SA2_F04** – The crates must be identifiable at the loading points via RFID technology **REQ_SA2_F07** – The platform should provide access to collected data **REQ_SA2_F08** – The platform should provide options to manage/view the connected mobile robots **REQ_SA2_F12** – The platform must be able to schedule carrier plans, based on real-time data **REQ_SA2_F13** – Mobile robots must be able to provide data to and receive control commands from the IoT-NGIN platform

	REQ_SA2_F14 – The platform UC app should allow the user to view the routes and carrier plans **REQ_SA2_F15** – The platform should allow the user to cancel or manage the carrier plans or routes **REQ_SA2_NF04** – IoT-NGIN should be scalable in terms of adding/removing devices and integrating hundreds of devices
Related KPIs	**KPI_SA2_03** – Reduction of the time needed for carrying crates to the loading point, compared to manual carrying >/= 10%
WP4 – IoT device access control	
Description	This module will provide access management to the AGLVs, before allowing access to monitoring data or control actions.
Adaptation and fine-tuning	No adaptation need identified at this point.
Deployment	Cloud
Related requirements	**REQ_SA2_F06** – The platform must ensure collected data sovereignty and integrity **REQ_SA2_F07** – The platform should provide access to collected data **REQ_SA2_F09** – The platform should provide options to manage users **REQ_SA2_NF01** – The IoT-NGIN platform must respect security and privacy requirements
Related KPIs	**KPI_SA2_01** – Collisions of mobile robots with human workers = 0 **KPI_SA2_02** – System reaction in emergency cases < 1 sec **KPI_SA2_03** – Reduction of the time needed for carrying crates to the loading point, compared to manual carrying >/= 10%

3.2.2.5 Testing Scenario

Table 21: Test scenario 1 "Obstacle avoidance".

Test 1: Obstacle avoidance	
Objective	The objective is to validate the effectiveness of the obstacle avoidance functionality of IoT-NGIN the Smart Agriculture LL UC5, in order to ensure labor staff safety and safe AGLV movement across the field while carrying crates with harvested crops.
Components	• IoT Device Access Control • IoT Device Indexing • MLaaS

	• Privacy Preserving FL • Secure Execution Framework
Features to be tested	• Obstacle detection • Route recalculation • Autonomous AGLV operation • Model training • Model deployment at the edge • Access control • Digital Twin of the AGLV
Requirements addressed	REQ_SA2_F01, REQ_SA2_F02, REQ_SA2_F03, REQ_SA2_F05, REQ_SA2_F06, REQ_SA2_F07, REQ_SA2_F10, REQ_SA2_F11, REQ_SA2_F13, REQ_SA2_F14, REQ_SA2_F15
Test setup	The IoT-NGIN framework should be deployed and functional. An AGLV is configured with the IoT-NGIN framework and able to communicate with the edge node. The AGLV must be integrated with IDI, in order to provide its data and receive commands. The AGLV App is installed on the AGLV, in which the ML-based obstacle avoidance service performs inference on the AGLV , based on collected data. The Smart Farmer must be registered on IoT-NGIN platform, in order to access AGLV data.
Steps	1. The user signs in the AGLV app. 2. The user configures a new route for the AGLV. 3. The AGLV starts its route. 4. Upon detection of obstacle, the AGLV should change its trajectory.
	5. The AGLV reaches its destination.
	6. The user accesses the route information.
KPIs	KPI_SA2_01, KPI_SA2_02, KPI_SA2_03
IoT-NGIN innovations	• Autonomous AGLV, integrated in IoT-Edge-Cloud continuum • Fast and easy over-the-air updates of AGLV intelligence • Shared intelligence building via privacy preserving FL training for object detection model, respecting and applying data sovereignty in Smart Agriculture

3.2.2.6 Use case sequence diagrams

The updated sequence diagram for UC5 is depicted in Figure 45, presenting the flow of operations and interactions among IoT-NGIN in order to realize the test scenario defined in the previous subsection.

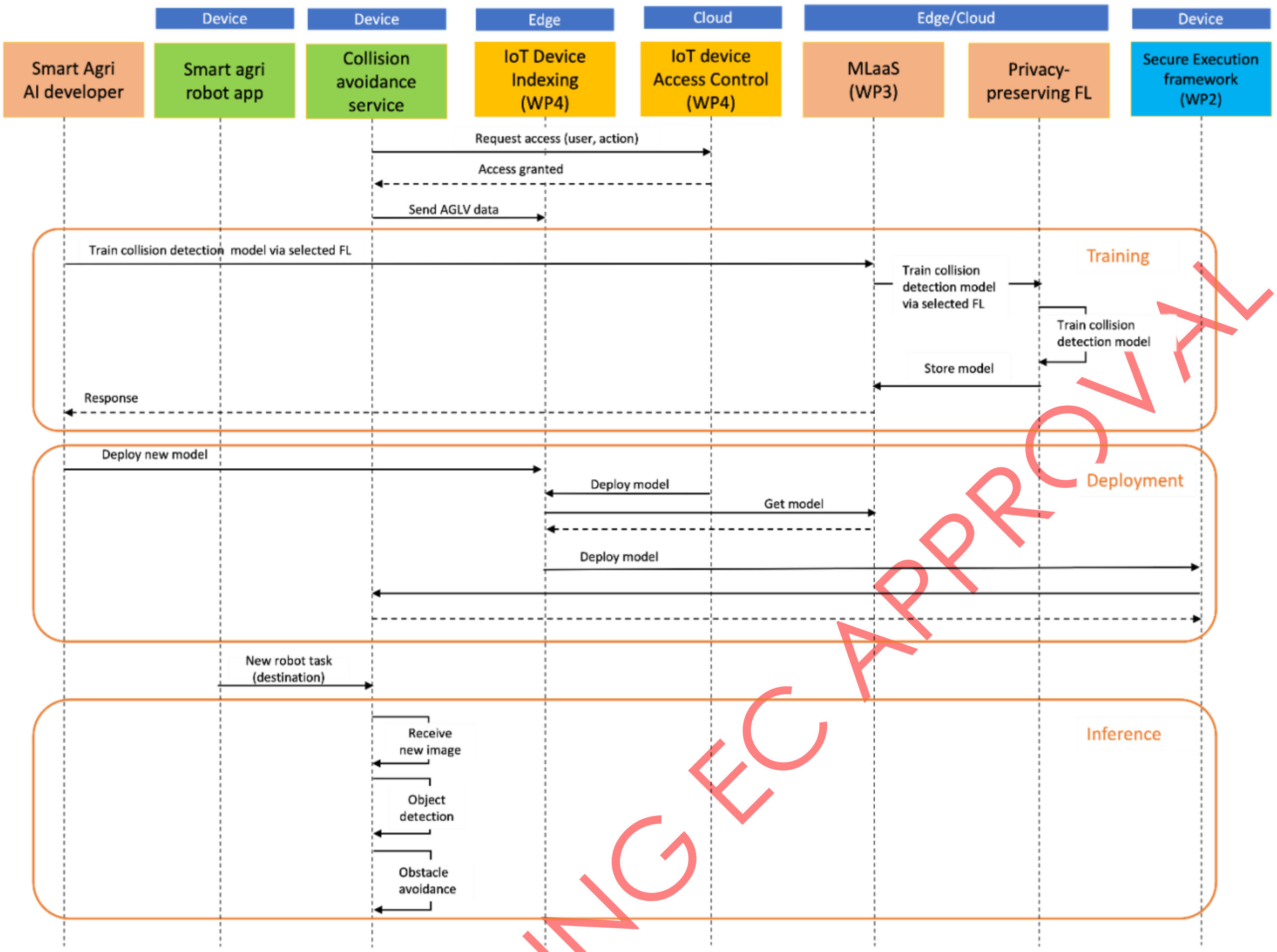


Figure 45: Sequence diagram for ML-based obstacle avoidance.

3.2.2.7 Execution timeline

The execution timeline of UC5, divided into multiple phases, is detailed in Table 22 below. The second phase has been successfully completed and we are ready for the last validation phase.

Table 22: Execution timeline of UC5.

Phase	Estimated start date	Estimated end date	Notes
Trial set-up and equipment procurement	M5	M18	Completed
Initial implementation and validation	M19	M30	Completed
Final implementation and validation	M31	M36	Validation mainly on the basis of the final integrated prototype

3.2.2.8 Intermediate Results

The pilot activities for UC5 so far have been focused on:

- Smart Agri AGLV prototype release, integrating on-device ML based object detection and collision avoidance, as well as autonomous movement
- Data collection through integration of IoT-NGIN's digital twinning functionality with the AGLV
- IoT-NGIN framework deployment and validation on the Smart Agriculture LL

For the needs of the IoT-NGIN validation under UC5, the IoT-NGIN tools already installed in the Smart Agriculture LL Kubernetes cluster for UC4 have been used. The IoT-NGIN components deployed on the LL and involved in the validation of this UC within this report include:

- IoT Device Indexing
- IoT Device Access Control
- Privacy-preserving FL & API

Moreover, the MLaaS platform deployment in OneLab is used for the delivery of ML services, integrating the Model Sharing component, used for storing the object detection ML models and retrieving them for deployment on the AGLV.

Moreover, the "Collision avoidance service" has been deployed on the AGLV and the "Smart Agri robot app" on the user's device (mobile phone).

In the following, the LL activities for each of the test scenarios under UC5 are briefly presented.

3.2.2.8.1 ML-based Object Detection

Within the context of this UC, we have performed finetuning of the obstacle detection algorithm in a federated manner in order to detect the class "person", as in UC5 the AGLV must detect and avoid human workers to ensure their safety. The training procedure is performed through the PPFL-API, described in D3.4 [20], with the NVIDIA-Flare framework [21]. NVIDIA-Flare has been selected as the preferred FL framework, since it has been implemented with real world clients and represents real-world circumstances. FL training is considered in two edge devices, i.e., with 2 clients in the FL system, in which each client has its own data. FL training has been configured for 3 rounds and 4 epochs in each round.

Initially, as shown in Figure 46, the FL system is started by sending a POST request to the PPFL-API. Then the server waits for each client to connect to the system. We use 2 different edge nodes, each of them being a real-world client in the FL system. After each client is connected to the system, an additional POST request is sent to the PPFL-API, as presented in Figure 48, with the FL parameters specified, in order to trigger an FL training task. It should be noted that the blurred parts of the image hide the credentials for the model storage component of the MLaaS platform, which is used for storing the final aggregated model and making it available later for deployment on the AGLV.

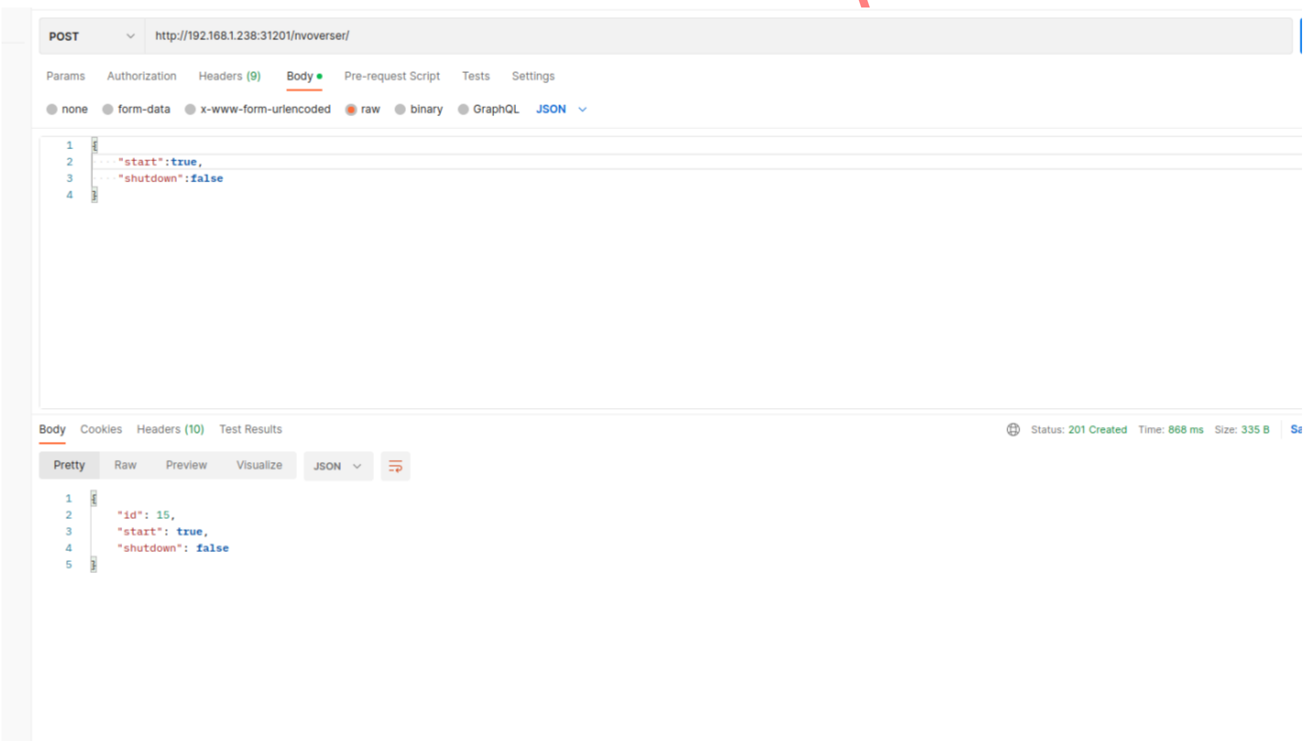


Figure 46: POST request in MLaaS PPFL-API which activates the FL system.

The relevant services have been indeed deployed on the LL K8S cluster, as depicted in Figure 47. Specifically, the first part depicts the PPFL API components running and ready to receive requests for FL training, while the second part of the figure show the components initiated after a request for starting the NVIDIA Flare system has been submitted to the PPFL API.

(a)

(b)

Figure 47: The PPFL API deployments in the Smart Agriculture LL K8s cluster (a) before and (b) after initiating NVIDIA Flare for training.

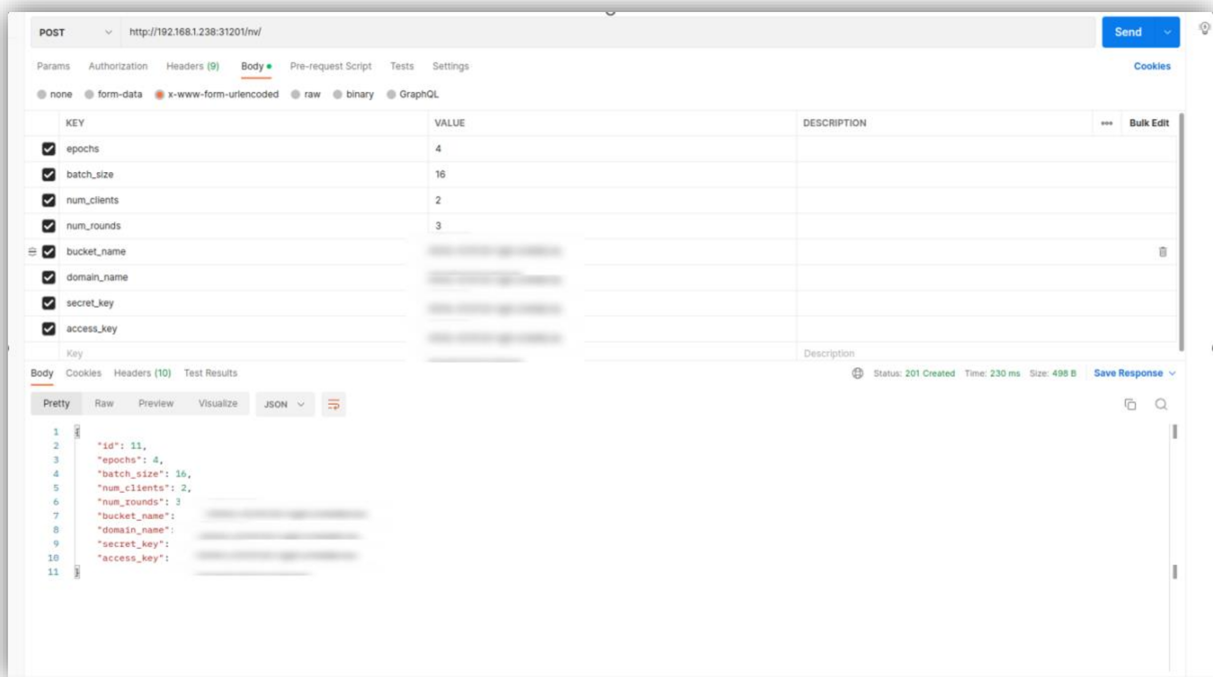


Figure 48: POST request in MLaaS PPFL-API for starting the FL training task.

When the training procedure is finished, the trained model is indeed stored in the MinIO object storage component of the MLaaS platform hosted in OneLab, as depicted in Figure 49. The trained model achieves 85.6% mAP (mean Average Precision) and thus it is considered of appropriate performance to detect successfully the class "person".

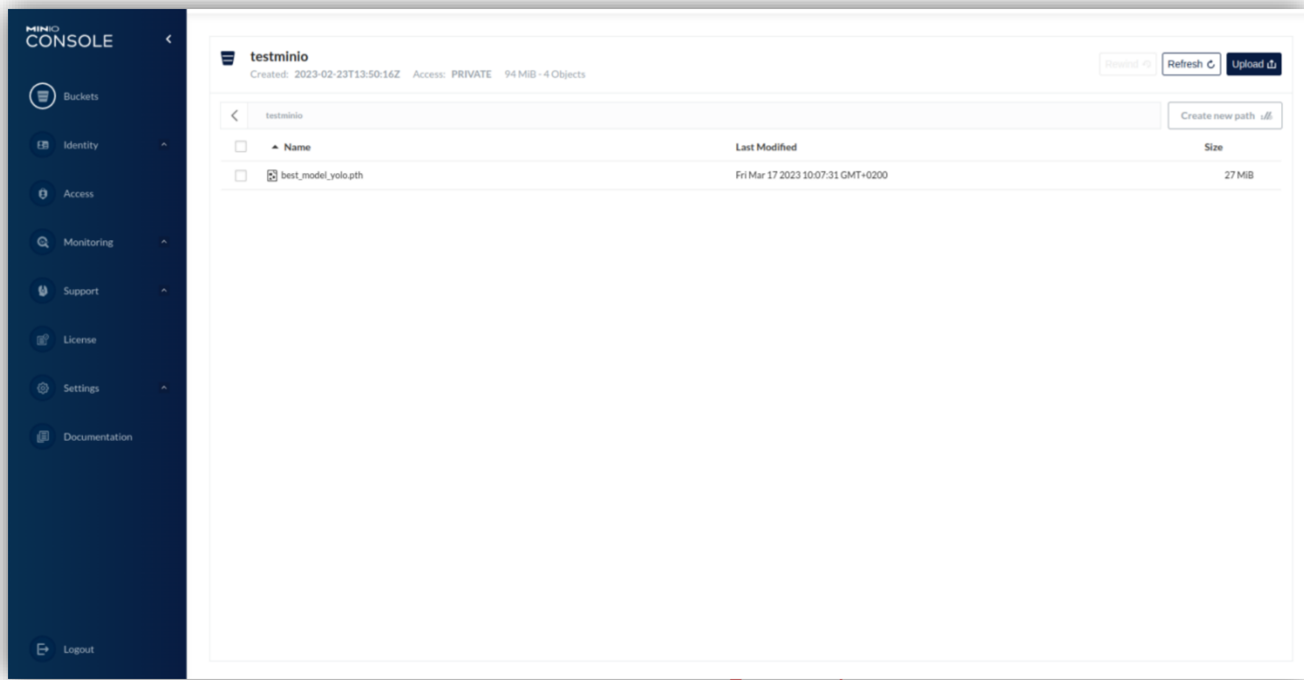


Figure 49: The aggregated model is stored in MLaaS storage.

3.2.2.8.2 Digital Twin-based actuation

After the object detection training is finished, the robot downloads the trained model from MLaaS model storage (MinIO) and uses it for the *Collision Avoidance* service. The download procedure is performed as a southbound communication of commands in IoT-NGIN's IoT-edge-cloud integration, as described in D6.2 [5]. As presented in Figure 50, the FIWARE-based IoT Device Indexing (IDI) component of IoT-NGIN sends a request to the AGLV with the model's URL in order to inform the AGLV that a new model exists in MinIO. Then, the AGLV downloads the new model from the provided URL.

```
 * Serving Flask app 'wrap' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
[17/03/2023 11:20:19 AM - INFO]: WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8080
 * Running on http://192.168.169.1:8080
[17/03/2023 11:20:19 AM - INFO]: Press CTRL+C to quit
[17/03/2023 11:20:19 AM - INFO]:  * Restarting with stat
[17/03/2023 11:20:20 AM - WARNING]:  * Debugger is active!
[17/03/2023 11:20:20 AM - INFO]:  * Debugger PIN: 102-252-852
[17/03/2023 11:20:31 AM - INFO]: Received model url 'minio-cli.kf.iot-ngin.onelab.eu' from FIWARE Orion!
[17/03/2023 11:20:39 AM - INFO]: New model 'best_model_yolo.pth' retrieved from MinIO!
[17/03/2023 11:20:40 AM - INFO]: Running new Yolov5 model..
[17/03/2023 11:20:40 AM - INFO]: 192.168.169.1 - - [17/Mar/2023 11:20:40] "POST /get_url HTTP/1.1" 200 -
```

Figure 50: The robot downloads the trained model from OneLab's MinIO object storage.

Afterwards, the *Collision Avoidance* service is executed using the newly downloaded Yolov5 trained model. During the collision avoidance procedure, the AGLV sends logging data to its Digital Twin residing on the edge device. Through this, the user may access information about detected obstacles, the number of avoided collisions etc.

The logging data are formalized into 2 categories, the *information* category and the *summary* category. The first category refers to information about the detected obstacles during robot's route. Particularly, when the robot detects an obstacle, it captures images before and after the avoidance and thus context is provided about AGLV's behavior during its route. These images are stored to the MinIO object storage, and the URL for each captured image that has been stored is sent to the Device Indexing, allowing to keep track of AGLV's behavior at each obstacle. Figure 51 shows a captured image with a detected "person" from the AGLV that has been stored to MLaaS MinIO. The URLs of the captured images are sent to IDI, after being granting access by the Access Control component. This is conducted via an *http* request, defining the special headers for implementing multitenancy, as detailed in D4.3 [22]. In this category, a header is specified as */info*, which represents the database where the images' URLs will be stored in the device's Digital Twin, and a specified sub-header as */url*, since the provided information refers to URLs. Figure 52 shows the output of the service that saves and sends the images to IDI.

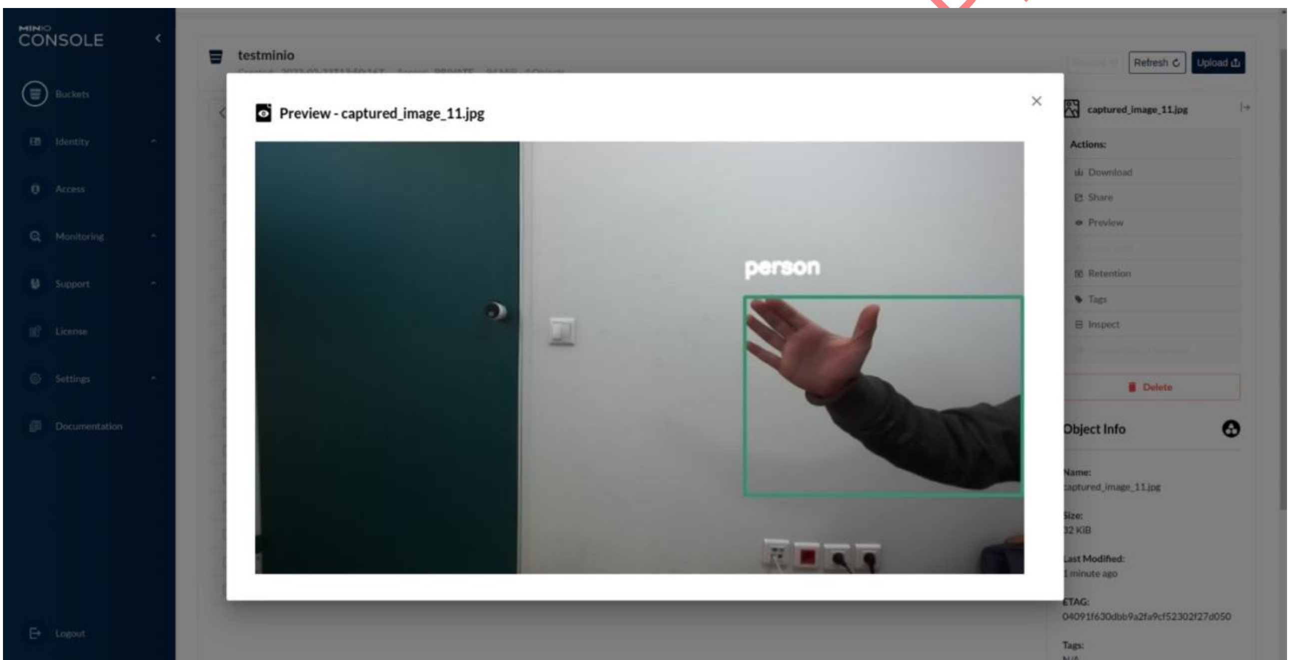


Figure 51: Image from AGLV that has been stored to OneLab's MinIO object storage. The URL for this image is then sent to the Device Indexing.

```
Image saved to MinIO bucket!!
-> Sending measurement: <Response [200]>

{}
Image saved to MinIO bucket!!
-> Sending measurement: <Response [200]>

{}
Image saved to MinIO bucket!!
-> Sending measurement: <Response [200]>

{}
Image saved to MinIO bucket!!
-> Sending measurement: <Response [200]>

{}
Image saved to MinIO bucket!!
-> Sending measurement: <Response [200]>
```

Figure 52: Saving images to the MLaaS MinIO and sending them to IDI.

Meanwhile, the logs from the *Collision Avoidance* service are also sent to the Device Indexing, providing context about AGLV's routes. The *Collision Avoidance* service outputs the coordinates of the goal point (i.e., the point that the AGLV must go to) and the direction of the detected obstacles with respect to the camera, e.g., front, front-left, front-right, front and front-left, front and front-right as shown in Figure 53. These additional logs provide a better understanding of AGLV's behavior and are sent to the IDI through an *http* request with the same header as the URLs (/*info*) and a specified sub-header as /*logs*.

```
root@jetson-xavier:/# rosrun motion_plan yolov5_nav.py

[INFO] [1679044157.469238]: Goal: (2, 0).
desired_yaw: -0.04046256618406779
current_yaw: -0.07114513669488022
[INFO] [1679044157.802804]: Yaw error: [0.030682570510812433]
[INFO] [1679044157.825938]: State changed to [1]
[INFO] [1679044157.874687]: Front right
[INFO] [1679044180.869888]: Front and Front-right
[INFO] [1679044180.937669]: Front right
[INFO] [1679044181.268583]: Front and Front-right
[INFO] [1679044181.328215]: Front right
[INFO] [1679044182.267167]: No obstacle
[INFO] [1679044182.338940]: Front right
```

Figure 53: Logs of our collision avoidance algorithm during its execution.

Regarding the *summary* category of logs, this allows keeping history of the main events of the AGLV routes such as the number of the detected obstacles and the number of the avoided ones, after the AGLV has reached its goal point, i.e., keeping a summary of the AGLV route. The summary is sent to IDI using an *http* request, similarly to the *information* category, with a specified header as /*summary* and a specified sub-header as /*logs*.

3.2.2.8.3 Processor – Microcontroller Connectivity

In this subsection, the hardware integration of the processor and the microcontroller is validated, ensuring that bidirectional communication of them is possible. This allows ROS-driven motion control commands to be passed to the microcontroller, as well as log information, possibly feedback originating from the motors, to be passed to the processor.

Each operation of the Jetson Xavier NX processor is taking place within the ROS framework. However, SynBoard must be able to read the generated commands from the Jetson Xavier NX processor. These generated commands are in a ROS topic and thus Synboard must activate a ROS node in order to subscribe to the corresponding topic and receive the generated commands. ROS provides the *rosserial_python* package [23], a package that allows ROS to communicate with hardware devices over a serial port using the PySerial library in Python. Serial communication is commonly used to communicate with microcontrollers, sensors, and other hardware devices in robotics and automation applications. The *rosserial_python* package provides a convenient way to integrate these devices with ROS, allowing data to be transmitted and received in a standardized ROS message format. The *rosserial_python* package can be launched as a ROS node, which will handle the communication between the serial device and ROS. The node can be configured to publish data to ROS topics, subscribe to ROS topics etc.

However, *rosserial_python* package has been implemented to work only in Arduino microcontrollers. We managed to adapt the package to our Synboard's specifications in order to work in our setup and run the ROS framework in Synboard. Figure 54 shows the successful connection of Jetson Xavier NX with Syboard. As shown in the figure, the Synboard is able to publish and subscribe from the topics of interest.

```
root@jetson-xavier:/# rosrun rosserial_python serial_node.py /dev/ttyUSB0 _baud:=9600
[INFO] [1679062502.674261]: ROS Serial Python Node
[INFO] [1679062502.685986]: Connecting to /dev/ttyUSB0 at 9600 baud
[INFO] [1679062504.798811]: Requesting topics...
[INFO] [1679062504.950773]: Note: publish buffer size is 512 bytes
[INFO] [1679062504.962654]: Setup publisher on pub_info [std_msgs/String]
[INFO] [1679062505.044652]: Setup publisher on rwheel_ticks [std_msgs/Int16]
[INFO] [1679062505.135381]: Setup publisher on lwheel_ticks [std_msgs/Int16]
[INFO] [1679062505.241237]: Note: subscribe buffer size is 512 bytes
[INFO] [1679062505.255135]: Setup subscriber on toggle_led [std_msgs/Empty]
[INFO] [1679062505.326946]: Setup subscriber on rmotor_cmd [std_msgs/Float32]
[INFO] [1679062505.416841]: Setup subscriber on lmotor_cmd [std_msgs/Float32]
```

Figure 54: Running the rosserial_python package to connect Jetson Xavier NX with Synboard.

3.3 Industry 4.0 Use Cases Living Lab

3.3.1 UC6 – Human-centered safety in a self-aware indoor factory environment

The collaboration of humans and robots in heavy industrial environments is increasingly common. As the number and speed of robots and automated machinery increases, productivity in factories is also rising, however, it exposes humans to new hazards. Moreover, the interactions between humans and machines are also becoming more frequent and complicated, requiring the development of new technologies such as Ambient Intelligence, Tactile IoT, and Augmented Reality to simplify and enhance these interactions.

This use case is divided into two parts to demonstrate different technologies developed in the IoT-Ngin project.

- **Part 1: Collision Prevention between Humans and Autonomous Guide Vehicles (AGVs)**
  This sub-use case aims to predict, identify, and prevent collisions between humans and AGVs as well as between Human Driven Vehicles (HDVs) and AGVs. The objective is to enhance the performance of human-driven vehicles by reducing the need for emergency braking, and to decrease the occurrence of scrap due to the falling of containers in emergency braking situations. Federated and interworked IoT technology will be utilized for localization, high-speed wireless communication, edge computing, and distributed AI for analyzing input from multiple sensors and cameras.
- **Part 2: Augmented Reality (AR) Human Assistance**
  This sub-use case involves assisting human workers in warehouses with the use of AR technology to recognize specific AGVs and receive relevant information. An AR interface will be deployed to provide information about the position of AGVs, HDVs, and humans, as well as to provide information about potential collisions identified in part 1.

3.3.1.1 Trial site description

This use case is being conducted in the Bosch factory at the city of Aranjuez (Spain). It is carried out by some engineers and operators of Bosch together with different researchers of i2cat with expertise in IoT, IA, 5G and AR technologies.

An initial description of the trial site was already provided in D7.2. It was supported by a sketch of the working area in which it was shown the existing elements (Load Station, ails and walls, working units, AGVs, etc.) and the new equipment needed to be installed for the use case (cameras, UWB beacons, LED lights, edge servers, etc.) At the current moment, all the equipment for the use case is installed and we can provide a more accurate description of the trial site.

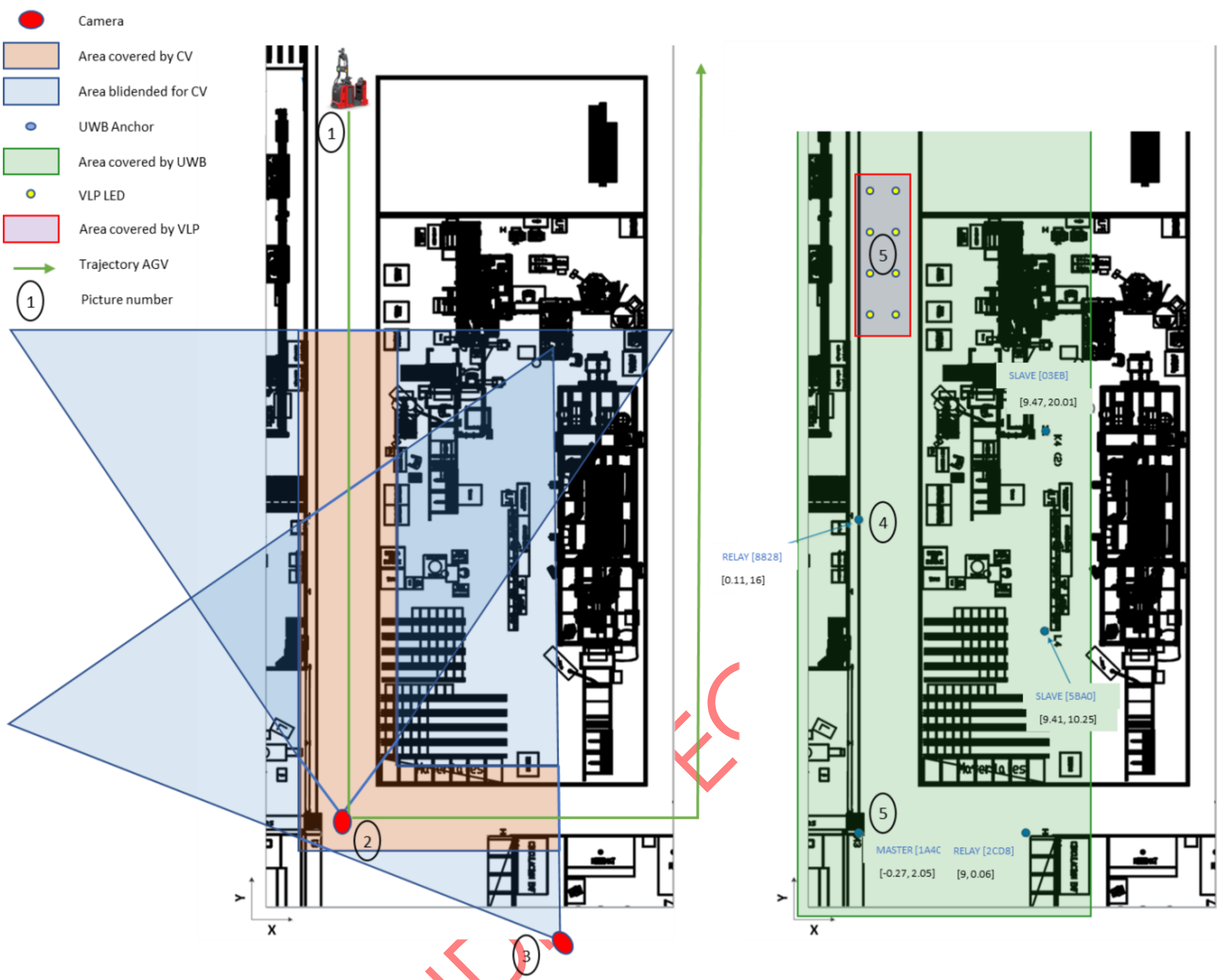
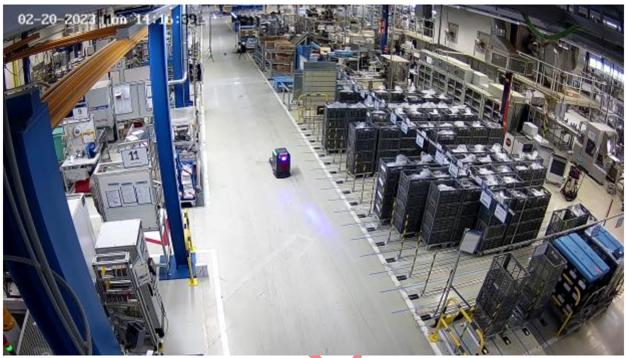
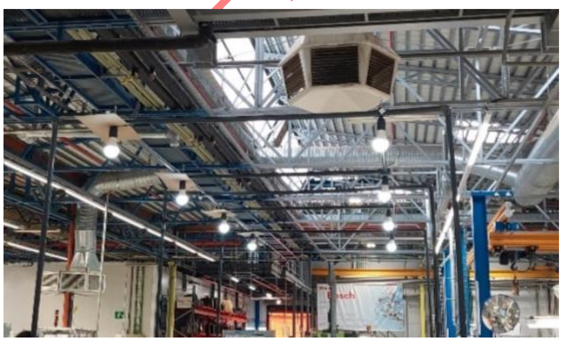


Figure 55: Trial site of UC6 in the BOSCH factory.

1 Autonomous guide vehicle 	2 View Camera 1 
3 View Camera 2 	4 UWB Anchor and Camera 1 
5 VLP LED grid 	5 VLP receiver (prototype mounted on low-cost robot, initial tests) 

5 UWB tag and VLP receiver installed in the AGV 	

Figure 56: Images of the trial site.

3.3.1.2 Used equipment

The following table describes the equipment already specified in D7.2maintained in D7.3. Additional description has been included.

Table 23: Equipment used in UC6.

Equipment	Description / Specifications	Type	Status
Autonomous Guided Vehicle	Robot CARY (Metralabs). Dim [mm]: 1040x402x656	Vehicle	Available
Screen	Laptop MSI Pulse GL66 11UEK-061XES Intel Core i7-11800H/16GB/1TB SSD/RTX 3060/15.6"	Hardware	Available
Camera	3 x Hikvision DS-2DE2A404IW-DE3-W 2-inch 4 MP 4X Powered by DarkFighter IR Network Speed Dome	Sensor	Available
UWB anchor	At least 4x Embedded devices used as a reference for indoor positioning, composed of a UWB transceiver and a computing unit.	Hardware/Software	Deployed in WP4. Adapted

	Deployed in the area. Currently, 5 units are deployed		for WP7
UWB tag	Embedded device used for indoor positioning, composed of a UWB transceiver and a computing unit.	Sensor/Software	Deployed in WP4. Adapted for WP7
VLP LEDs	Light infrastructure used for Visual Light Positioning. Composed of commercial LEDs with the capability to switch the LEDs and transmit the information to allow positioning. Also, a Wi-Fi communication interface is deployed to allow remote configuration/tuning.	Hardware/Software	Deployed in WP4. Adapted for WP7
Edge server PC	Intel Core i9-12900F 2.4 GHz, DDRS 4800MHz 32 GB CL40	Hardware	Available
GPU	A Nvidia RTX 3080 Ti GPU	Hardware	Available
VLP receiver	Embedded system composed of a camera, a computing unit and wireless communications to transmit the position in real-time.	Hardware	Being deployed in WP4. Will be adapted for WP7
Forklift	Still *TBD*	Vehicle	To be procured
Wi-Fi AP	Wi-Fi AP to provided local connectivity. The hardware also provides cellular connectivity for external access.	Hardware	To be procured

Also, new equipment requirements have been added to the trial site.

Table 24: New equipment required in UC6.

Equipment	Description / Specifications	Type	Status
Switch POE	5 ports Ethernet switch with POE to connect and power the cameras	Hardware	Available
New AGVs	A Tugger AGV in addition to the Robot CARY	Vehicle	Available

The herein mentioned table includes the details of the equipment that has been modified from what was described in D7.2.

Table 25: Details of equipment in UC6.

Equipment	Old	New	Reason of modification
Camera	3 x Hikvision 2-inch 4 MP	2 x Hikvision 2-inch 4 MP	It is possible to cover the intersection and the aisle with 2 cameras
Jetson Nano		Not needed	We will process the video from the 2 cameras in the edge server with a Nvidia RTX 3080 Ti GPU

3.3.1.3 Data collection

The datasets produced in this use case were already described in the D7.1 (Data Management Plant) and in D7.2. Although, once the measuring campaigns have started and intermediary results presented, it is possible to provide more precise information about the amount of data generated and also to explain the differences with the datasets planned in previous deliverables.

Table 26: Dataset for UC6.

Dataset	Data amount	Time span	Differences / Comments
Indoor surveillance camera videos	30 HD videos lasting from 10 seconds to 2 minutes. 536 Gb	30 frames per second	Contents personal data, it needs to follow the data privacy policy defined in D10.1
UWB raw data	Distance measurements from the UWB anchors registered by the UWB Tag. The total		
Indoor trajectories	+50 trajectories of different objects/workers resulting from CV-based positioning. AGV trajectory coming from UWB-based and CV-based positioning.	0.1s (CV,VLP), 1s (UWB)	D7.2 differentiated between Workers' and AGV trajectories. Now, the dataset is unified. Specific fields are used to identify the technology (VLP, CV, UWB) and the object (e.g. worker/AGV)

3.3.1.4 Alignment with IoT-NGIN technologies update

There is not any update from the previous deliverable D7.2 about the alignment with IoT-NGIN technologies for his use case.

3.3.1.5 Use case sequence diagram(s)

The sequence diagram Part 1 of the use case "**AGV – Human collisions prevention**" has not had any modification since the previous deliverable D7.2. On the other hand, a more detailed sequence diagram of the Part 2 of the use case "**AR human assistance"** is depicted in the following figure.

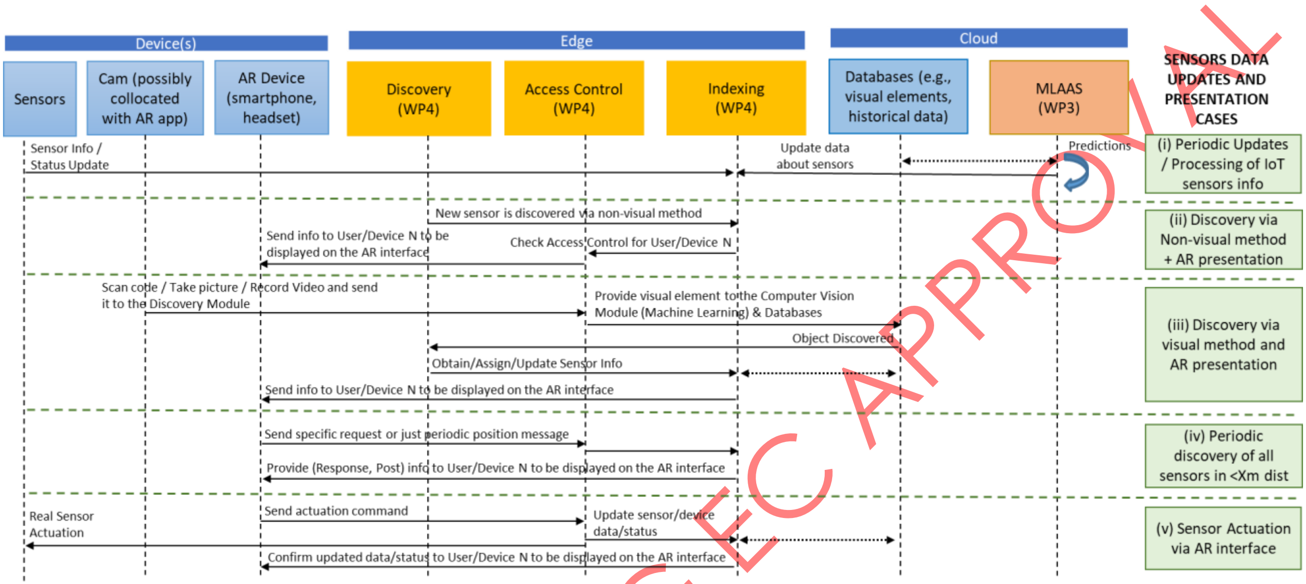


Figure 57: Diagram of UC6.

3.3.1.6 Testing Scenarios

The testing and validation procedures of the scenarios were already explained in section 3.3.1.6 of the deliverable D7.2. In the section of "Intermediate results" of this use case, the results of the test and validation of the IoT Device Discovery module based on Ultra Wide Band, Visual Light Positioning and Computer Vision are presented. The validation of the IoT Device Indexing Module, the Collision Detection Module, the AR interaction and the 5G network slicing will be presented in the following deliverable, D7.4 "IoT-NGIN Living Labs use cases assessment & replication guidelines".

3.3.1.7 Execution timeline

Table 27: Execution timeline of UC6.

Phase	Start date	End date	Notes
Trial set-up and equipment procurement	M9	M12 (March 2022)	- Trial scenario discussion and validation - Sensors, anchors and powering defined - Server requirements and connectivity defined - Equipment procurement - Initial test measurements in lab performed
Initial implementation and validation	M13	M21 (June 2022)	- Sensors deployment in the factory - AGVs routes and area covered defined - 1st Campaign of measurements in the factory - Objects position measurements tested and validated
1st On-site test in Bosch Factory in Aranjuez – June 2022			
Intermediate implementation and validation – Part 1	M22	M30 (March 2023)	- Indexing module tested in lab - Collision detection module deployment - Sensors installed in final location - 2nd Campaign of measurements in the factory
2nd On-site test in Bosch Factory in Aranjuez – March 2023			
Intermediate implementation and validation – Part 2	M31	M32 (June 2023)	- AR interaction tested in lab - 5G connectivity tested in lab - Wi-Fi AP and server installation in the factory - Sensors connected and powered and connected to the server - Test of the IoT device discovery in real time - Test of collision detection module in real time
3th On-site test in Bosch Factory in Aranjuez – June 2023			
Final implementation and validation	M33	M36 (September 2023)	- Test of the AR interaction component - Full demo validated
4th On-site test in Bosch Factory in Aranjuez – September 2023			

3.3.1.8 Intermediate results

This section summarizes the intermediate results obtained during the 1st and 2nd trial tests performed at Bosch factory in Aranjuez.

The validation of three modules of the IoT-NGIN project was performed:

- UWB-based IoT Device Discovery
- VLP-based IoT Device Discovery
- Computer vision-based IoT Device Discovery

The details of these tests are described below.

On the other hand, the validation of the following modules will be reported in D7.4 after the last on-site tests in the factory and in the lab (for the case of 5G network slicing)

- IoT Device Indexing
- Collision detection module
- AR interaction
- 5G network slicing

3.3.1.8.1 UWB-based IoT Device Discovery

Following the initial work reported in D7.2 the validation of the UWB-based IoT Device Discovery has been performed as follows:

- A set of 5 UWB anchors (with hardware and firmware enhancements from the versions tested in D7.2) have been deployed in the factory area. These anchors, powered by the electrical supply, have been installed at heights between 2.5 and 4 meters.

- Two UWB tag devices, powered by batteries, have been used to test and validate the tracking capabilities and the positioning accuracy for static and mobile conditions. For mobility, walking experiments and, also, tests with the UWB tags installed in the AGV have been performed.

The performed tests consisted in evaluating the UWB localization level and tracking system inside the AGV loading/unloading area of the Bosch's factory. As can be observed in Table 28, up to 5 anchors were deployed to cover part of the routes of the AGV.

In the next table, the identification, role, and position of each deployed anchor is shown.

Table 28: Position of anchors.

ID	Role	Position	
		X [m]	Y [m]
1A40	MASTER	-0.27	2.05
2CDB	RELAY	9	0.06
5BA0	SLAVE	9.41	10.25
8828	RELAY	0.11	16
3EB	SLAVE	9.47	20.01

As for the heights of the anchors, those were deployed at heights between 2.5 and 4, as it can be observed in the figures below.

Figure 58: Anchor master 1A40.

Figure 59: Anchor 2CDB.

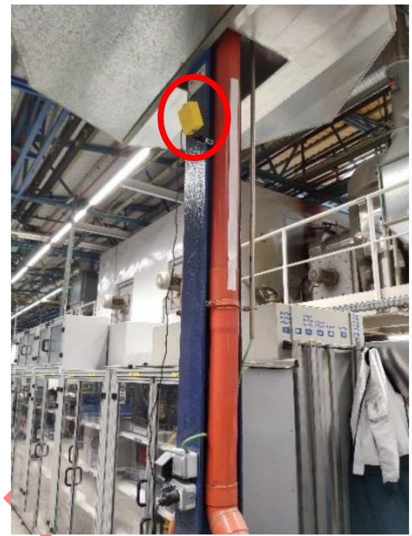


Figure 60: Anchor 5BA0.

Figure 61: Anchor 8828.

Figure 62: Anchor 3EB.

Herein, some results of the tests are provided:

3.3.1.8.1.1 Static positioning

In the tests, the UWB tag has been installed in a tripod at a height of 2 meters. The tripod has been statically located at different well-known positions for 1 minute. The system has reported one measurement each 1 second.

The following image shows the results obtained in the tests. As it can be seen the deviation of the estimated position in all the tests is small and the values are close to the real locations.

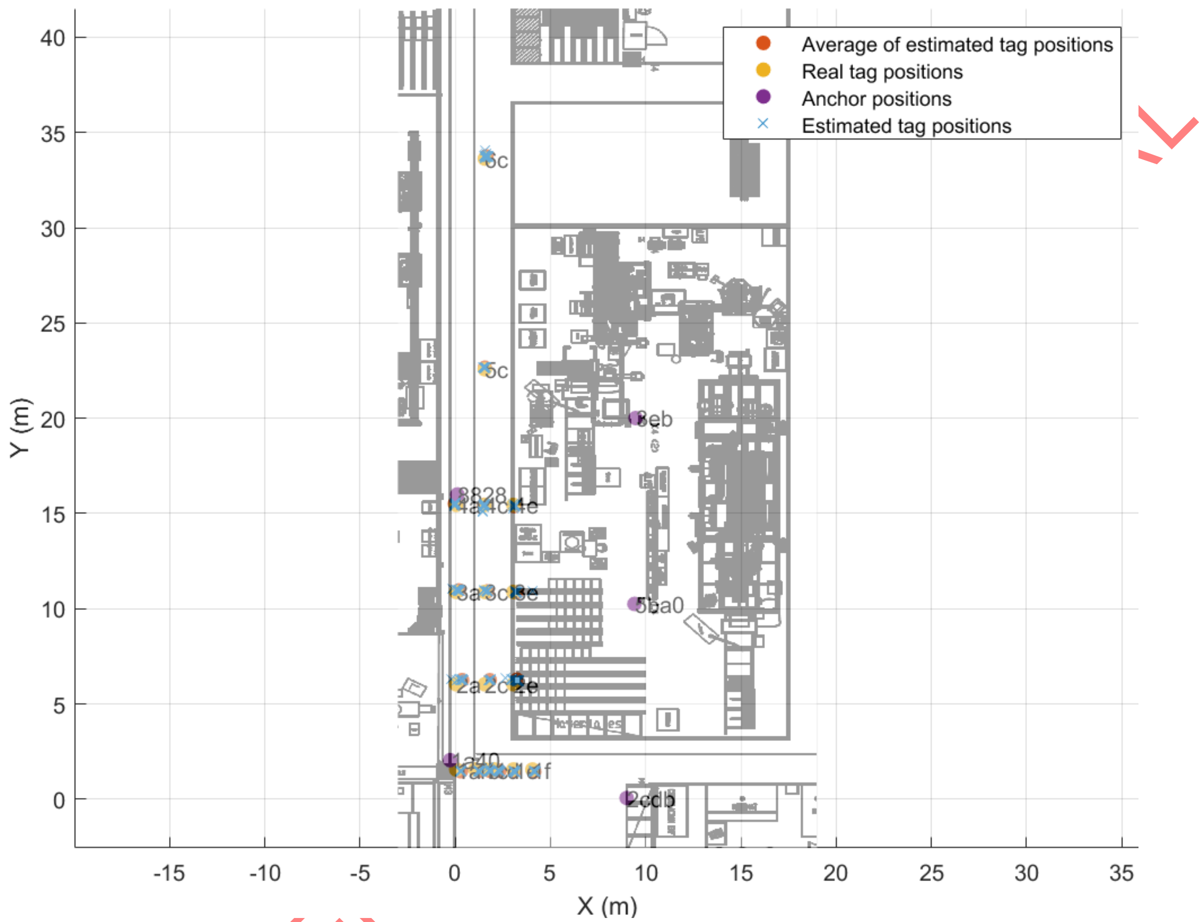


Figure 63: Positioning results. Static positions.

The following table summarizes the results for the static locations.

Table 29: Results of the static locations.

Positions	Real tag position		Nr of samples	Average of estimated tag positions		Standard deviation of estimated tag positions		2D-Error (cm)
	X (m)	Y (m)		X (m)	Y (m)	X (cm)	Y (cm)	
Pos #1A	0.05	1.56	52	0.30	1.46	2.22	1.45	26.50
Pos #1B	1.05	1.56	48	1.24	1.48	2.58	2.25	21.01
Pos #1C	1.55	1.56	74	1.80	1.50	2.76	3.06	25.93
Pos #1D	2.05	1.56	71	2.32	1.49	2.40	1.90	27.84
Pos #1E	3.05	1.56	83	3.09	1.47	2.70	1.74	10.16
Pos #1F	4.05	1.56	63	4.15	1.45	1.97	1.04	14.32
Pos #2A	0.05	6.03	77	0.36	6.25	8.74	1.71	39.23
Pos #2C	1.55	6.03	72	1.81	6.25	2.36	1.10	34.18
Pos #2E	3.05	6.03	87	3.24	6.28	9.72	2.55	33.21
Pos #3A	0.05	10.88	80	0.19	10.96	5.57	2.02	16.67
Pos #3C	1.55	10.88	73	1.67	10.91	5.09	1.95	13.37
Pos #3E	3.05	10.88	77	3.22	10.86	2.02	2.16	26.84
Pos #4A	0.05	15.45	97	-0.02	15.52	1.57	2.88	10.33
Pos #4C	1.55	15.45	82	1.51	15.45	1.76	6.27	5.25
Pos #4E	3.05	15.45	73	3.10	15.39	2.00	3.50	8.59
Pos #5C	1.55	22.57	71	1.55	22.65	1.98	1.50	8.59
Pos #6C	1.55	33.63	68	1.63	33.75	4.46	5.25	14.96

In most positions, the 2D error is below 30 centimeters, reaching errors of less than 15 centimeters in the best cases.

3.3.1.8.1.2 Mobile tests (walking speed)

In this test, the UWB tag installed in the tripod has been carried along a path surrounding the machines area, as shown in the graph. The intention of the test was to evaluate the performance of the system in Non-Line of Sight situations between the moving device and the deployed anchors (out of the covered area).

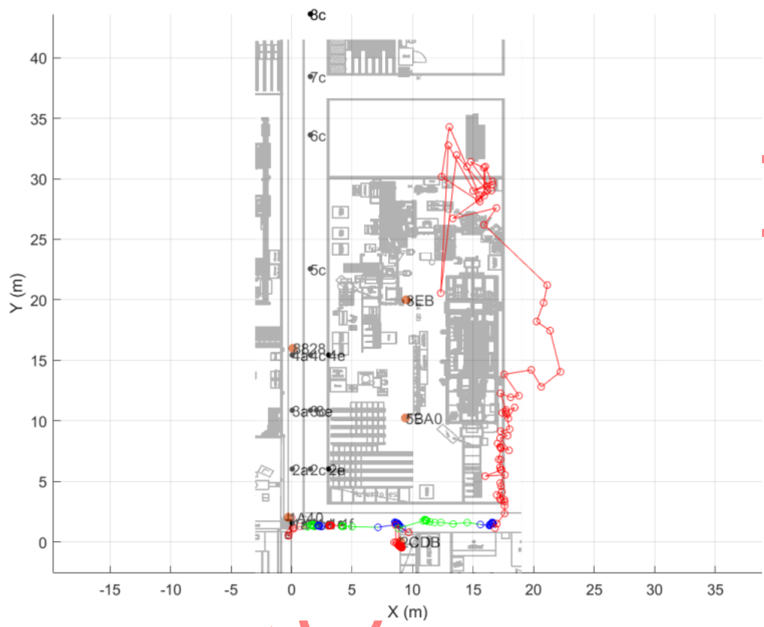


Figure 64: Walking path.

As it can be observed, the system performs well in the initial part (moving from bottom left to bottom right corners. Larger dots correspond to several stops in the path, where multiple position measurements are estimated. The behavior is similar to the one observed in the static tests. In the left side (depicted in red), the real trajectory can be still recognized for a while but when the NLOS situation gets worse, though UWB ranging are still received, the positioning error increases notably as expected.

3.3.1.8.1.3 Mobile tests (AGV speed)

In these tests two UWB tags have been installed in the AGV as shown in Figure 65. One of them was located in the front of the AGV and the second one was installed in the side. The aim of the tests is to determine the best location for the final deployment and to test the impact of the AGV structure in the measurements.

Figure 65: Tags UWB in AGV.

The following figure shows the results of the tracking of the AGV while it was moving. As it can be seen the estimated positions obtained by each of the tags allow to trace the path followed by the AGV. Note that, although the distance between the two paths varies along the way, the red and blue lines do not cross or overlap and remain on the side of the corresponding tag. From the tests it can be concluded that, though the differences are not very significative, the UWB tag depicted in red provides more stable measurements if compared to the route followed by the AGV.

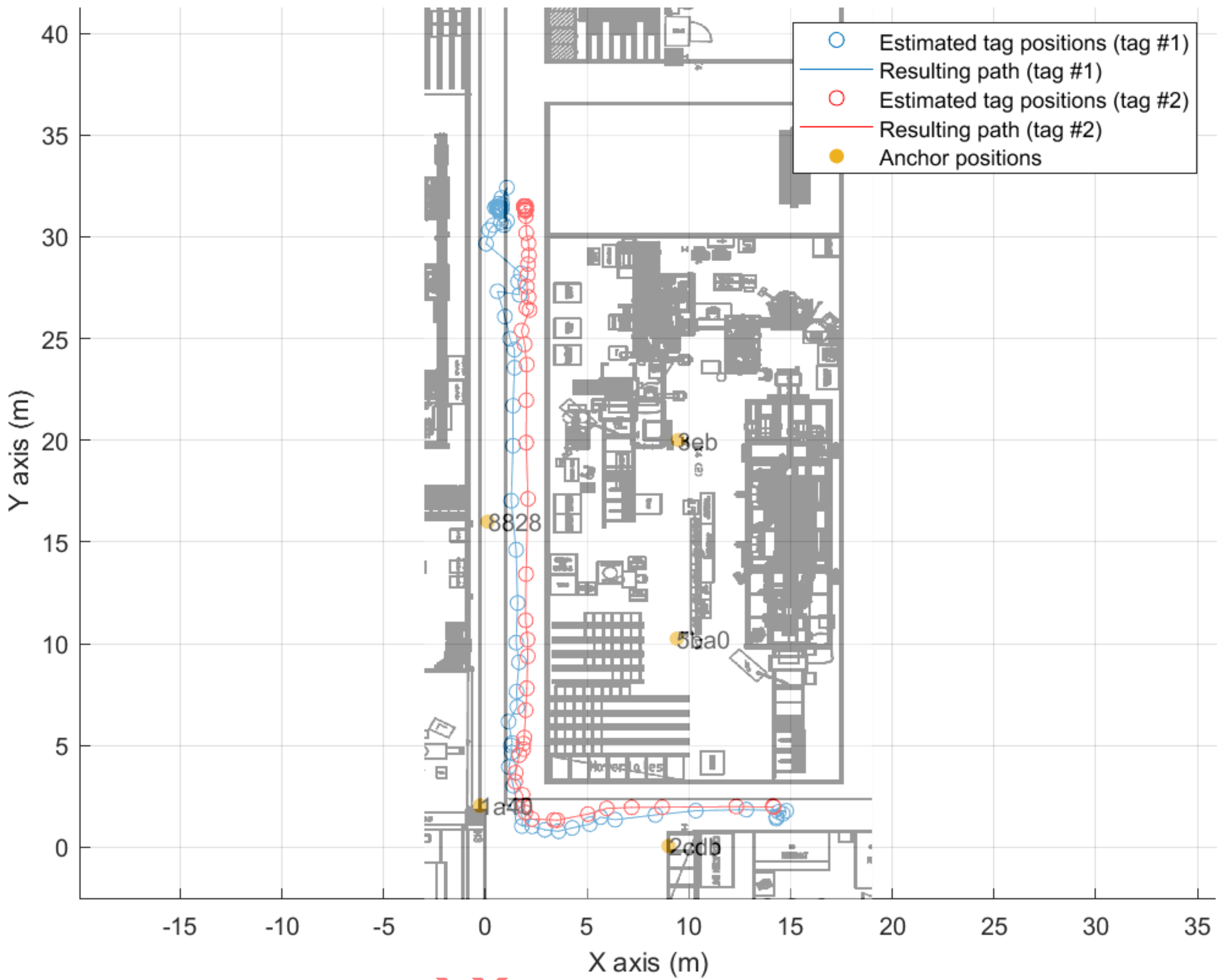


Figure 66: Mobile test with AGV results.

To conclude, the herein presented intermediate tests have served to validate the final installation of the UWB infrastructure at Bosch factory and to evaluate the performance of the IoT Device Discovery solution.

3.3.1.8.2 VLP-based IoT Device Discovery

To validate the VLP-based IoT Device Discovery an initial VLP grid infrastructure with 8 LEDs has been deployed at Bosch factory. The infrastructure was mounted a at height of approximately 2.5 meters using tripods. The dimensions of the infrastructure are: 1.2m x 3m (4.32 square meters). This deployment allowed to validate the system in a controlled scenario and to determine the final location of the installation.

The VLP receiver prototype uses Optical Camera Communications (OCC) technology to collect the information sent by the VLP infrastructure and to estimate its position. The initial prototype has been mounted on a line-follower robot to facilitate the analysis of the performance of the system. The solution has been also tested at the final location, installed on the AGV device.

Herein the results of the intermediate tests are summarized.

1. VLP prototype accuracy

As shown in the following image, a curved line is depicted in the ground and the robot has been used to follow it while the VLP IoT Device Discovery system is running.

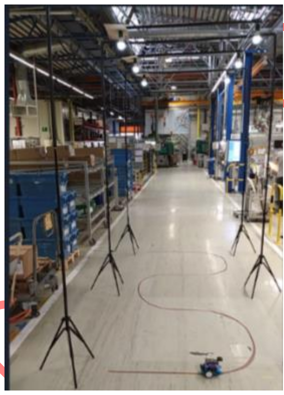


Figure 67: VLP tests. Path followed by the test robot.

Next graph shows the resulting positioning information once data is processed by the VLP receiver. Colors indicate the number of LED sources that have been used to estimate the position of the robot and cyan dots indicate the position of the VLP LEDs. As it can be seen, the curved trail can be recognized and follows the same shape as the original path. In few spots the system was not able to calculate a position. In general, new locations could be correctly estimated each 12 ms. Currently, the receiver sends position updates (aggregated values) through WiFI to the IoTNGIN platform each 500 milliseconds. The update rate will be adjusted in the final implementation to cope with the needs of the use case.

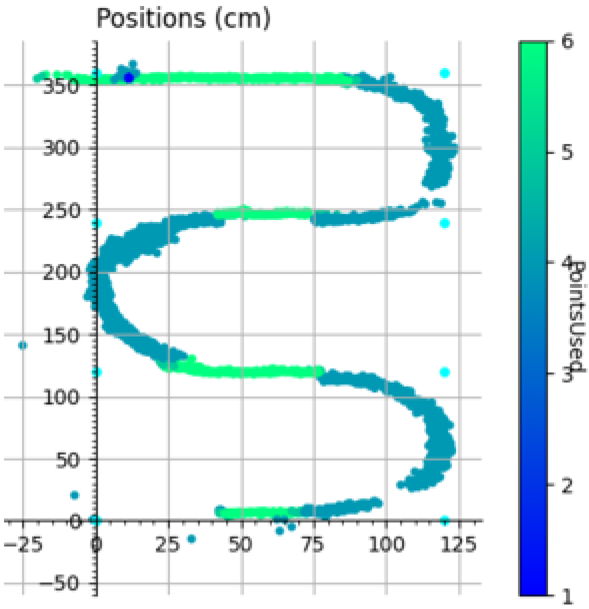


Figure 68: VLP tests. Estimated position along the path.

In these initial tests, the dispersion of the estimated positions was of 7.5 centimeters. As observed, the followed path can be clearly identified.

3.3.1.8.2.1 AGV positioning with VLP

The VLP receiver has been also mounted on the AGV to get some initial results of the performance of the system in the final deployment. A laptop PC has been used to collect local logs of the system that can help to better analyze the results and lately optimize the performance of the deployed algorithms.

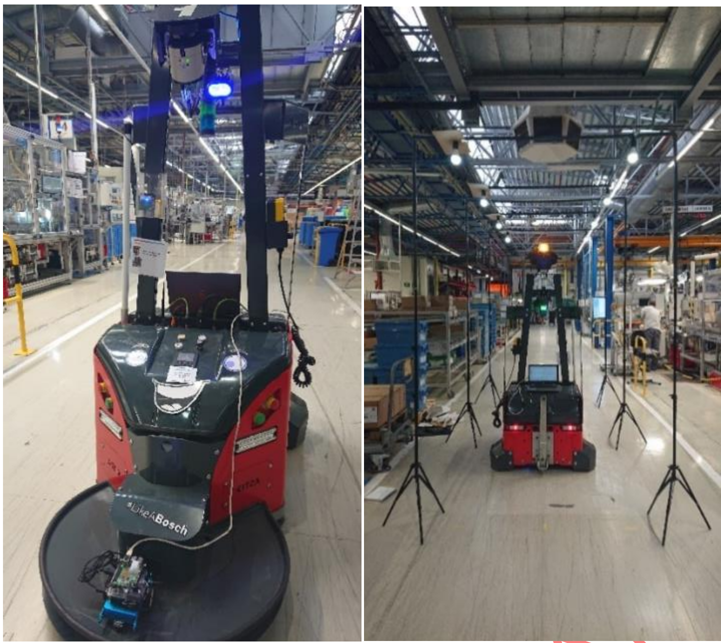


Figure 69: VLP system mounted on AGV.

The following graph shows the estimated positioning information once data is processed by the VLP receiver. In this specific test, the robot was manually controlled to traverse the VLP infrastructure in a straight line. It can be observed that at the start, in the middle and at the end the positioning is only possible using 4 LEDs as the AGV structure obstructs the vision of some of the lights. When 6 VLP LEDs are used, the accuracy of the system is improved.

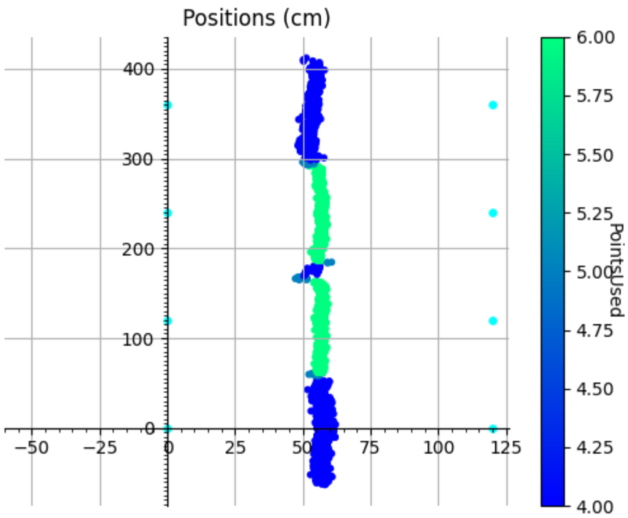


Figure 70: VLP tests. Straight path followed by the AGV.

To conclude, the tests have served to determine the final installation of the VLP infrastructure and the receiver and to obtain some initial results of the system working in the use case scenario. Currently, the VLP infrastructure is being installed in the ceiling of the factory site at a height of 4.2 meters. Once finished, more detailed tests will be performed.

3.3.1.8.3 Computer Vision-based IoT Device Discovery

Finally, the IoT Device Discovery module based on Computer Vision developed in WP4 was also tested and validated in the UC6.

The area chosen for this validation is shown in Table 30, it includes two aisles of the factory intersecting at 90 degrees. The wide consider for both ails is 4 m and the length is 20 and 50 m respectively.

In this test, three different types of objects were detected in the validation area, which are: Tugger AGV, loaded carts and persons. The metrics evaluated for the detection validation of each object is summarized in the following table.

Table 30: Metrics for the detection validation for each object.

Object	Precision	Recall	mAP	Minimum object size detected (pixels)
Tugger AGV	0.975	1	0.995	No minimum reached inside the working area
Loaded Cart	0.99	1	0.995	*idem*
Person	0.99	1	0.995	*idem*

The positioning of the Tugger AGV given by the IoT Device Discovery module was also validated. I was compared against 5 known positions of the AGV obtained by manual measurement. This comparison is shown in the following table. It can be seen that the error ranges from 0.16 to 0.75 m and that it increases when the Y distance also increases. This is due that higher distances in the Y axis are further away from the cameras installed next to the zero reference.

Table 31: IT discovery vs manual measures.

Positions	Real position		Estimated position by CV		2D-Error (m)
	X (m)	Y (m)	X (m)	Y (m)	
Pos #1	2.33	11.60	2.44	10.86	0.75
Pos #2	2.10	7.00	2.14	6.48	0.52
Pos #3	2.10	6.29	2.05	5.90	0.39

Pos #4	1.75	4.06	1.87	4.20	0.16
Pos #5	6.75	3.11	6.99	3.12	0.24

Finally, the tracking of the objects was also validated. The videos from the first two measurements campaigns have shown that the tracking worked for all the objects detected. The test was carried out also under circumstances in which the objects partially occluded between them and with other objects of the scenario (i.e.: columns or beams of the factory).

In the next figure it can be seen a snapshot of the detection, positioning and tracking of a person, and AGV and 5 loaded carts. The image on the left shows the detection of the 7 objects from the point of view of the camera, and the image of the right plots the detection and tracking of all the objects in a scaled map.

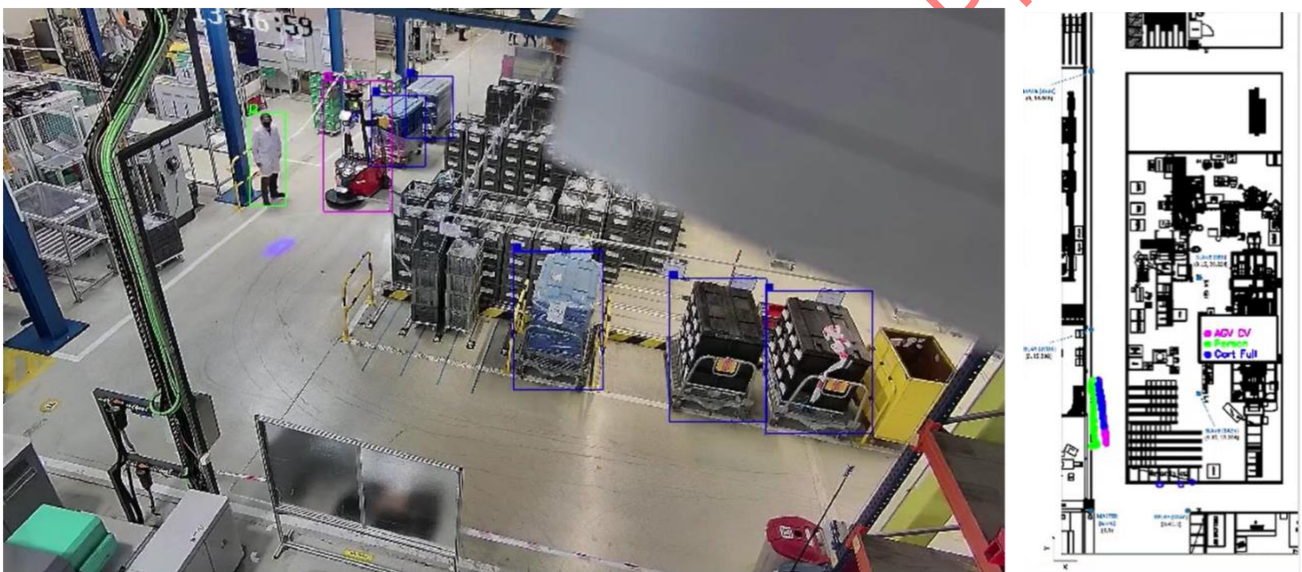


Figure 71: Simultaneous tracking of a person, an AGV and 5 loaded carts.

3.3.2 UC7 – Human-centred Augmented Reality assisted build-to-order assembly

The use case will be located at ABB's facilities and involves the assembly and wiring of ABB's drive cabinet products.

Digital models of the cabinets, containing both mechanical and electronic CAD data are developed. These digital cabinets are used to visualize the different assembly phases to the assembly worker in the proprietary Smart Wiring™ software created by EPLAN.

The same digital models are used to create an AR application. Initially AR application was planned for training, sales, manufacturing and/or maintenance purposes. All the potential use cases were investigated and the maintenance-related was selected to proceed with in this use case.

3.3.2.1 Trial site description

The use case will take place at ABB's factory in Helsinki on the cabinet production line. The main UC component is the digital cabinet model created by combining MCAD data (generated by Creo™ software) and ECAD data (generated by EPLAN's software suite). The digital model is then used to visualize assembly steps in EPLAN's Smart Wiring™ software.

Second UC component is the AR application for supporting maintenance operations. The application helps the operator to locate electrical components of a drive and brings details of the components for the operator. The application is created utilizing the repository of software tools provided by T4.4. AR application will be tested in a controlled environment with a physical cabinet.

3.3.2.2 Required equipment

Table 32 below shows a detailed description of the required equipment for UC7, along with its type and status (whether it is available or still to be procured). It was decided to use cloud server for AR component, because of its more flexible availability in remote locations.

Table 32: Detailed description of the required equipment for UC7.

Equipment	Description / Specifications	Type	Status
Cloud Server	Capable of running containers / Kubernetes and NGIN components.	Server	Available
Mobile device	Mobile phone / tablet with camera. Capable of running the AR app.	End device	Available
AR IDE / SDK	A framework for developing the AR application. Should have tools for importing CAD models.	Software	Available

Smart Wiring™	Proprietary system tracking the wiring status during assembly.	Software	Available

3.3.2.3 Data collection

The following dataset has been identified as part of the UC:

- **CAD data:** digital model of the drive cabinet that will be used to visualize assembly steps in the Smart Wiring™ software and in AR application development.
- **Electrical equipment list:** IDs and details of the electrical devices contained in the cabinet. Their physical location is highlighted in the app per user selection.
- **Wiring job progress:** User marks wires installed with the Smart Wiring ™ application. Progress is recorded to a local server.

3.3.2.4 Alignment with IoT-NGIN technologies

Table below shows the IoT-NGIN technologies relevant for UC7. For each technology, a description of its role is provided along with any adaptation it might need to be used in the UC, deployment details and related UC requirements and KPIs. Requirement REQ_IN2_F08 has been modified and requirements REQ_IN2_F05 and REQ_IN2_F11 excluded due to refinement of the AR component of the UC.

Table 33. Detailed description of the required equipment for UC8.

WP4 – IoT device discovery – Non-visual (QR code)	
Description	Cabinets are equipped with QR-codes which can be scanned using a mobile app to identify the specific cabinet type and fetch the relevant information from a server.
Adaptation and fine-tuning	N/A
Deployment	Deployed on a private server.
Related requirements	**REQ_IN2_F02** – The application must provide information through touch screens and public displays on the shop floor. **REQ_IN2_F03** – The application requires a GUI that consistently displays the relevant data and adapts to the current state. **REQ_IN2_F12** – The user can select different views, process data, and charts to be displayed. **REQ_IN2_NF05** – No personal data is gathered or processed, so that it is not possible to identify any person on the shop floor
Related KPIs	N/A

WP4 – IoT AR toolkit	
Description	The AR software & tools provided by task 4.4 will be used to create an AR application which utilizes the digital model of the cabinet. The AR application can manipulate the digital model and can be used for training, sales and/or maintenance purposes.
Adaptation and fine-tuning	Need for CAD-model importing. The CAD model developed in the use case needs to be imported into the AR application development environment. E.G. Unity, Unreal Engine, and Vuforia Studio have existing proprietary tools for importing CAD-models.
Deployment	The AR application will run on mobile devices as an app.
Related requirements	**REQ_IN2_F03** – The application needs a GUI that always shows the relevant data and adjusts itself to the current state **REQ_IN2_F08** – The modalities of the service will be training and guiding maintenance operators. **REQ_IN2_NF05** – No personal data is gathered or processed, so that it is not possible to identify any person on the shop floor
Related KPIs	N/A

3.3.2.5 Use case sequence diagrams

Figure 72 below shows a refined planned sequence diagram of UC7, depicting the involved actors in the UC and their interactions with the relevant IoT-NGIN technologies and components.

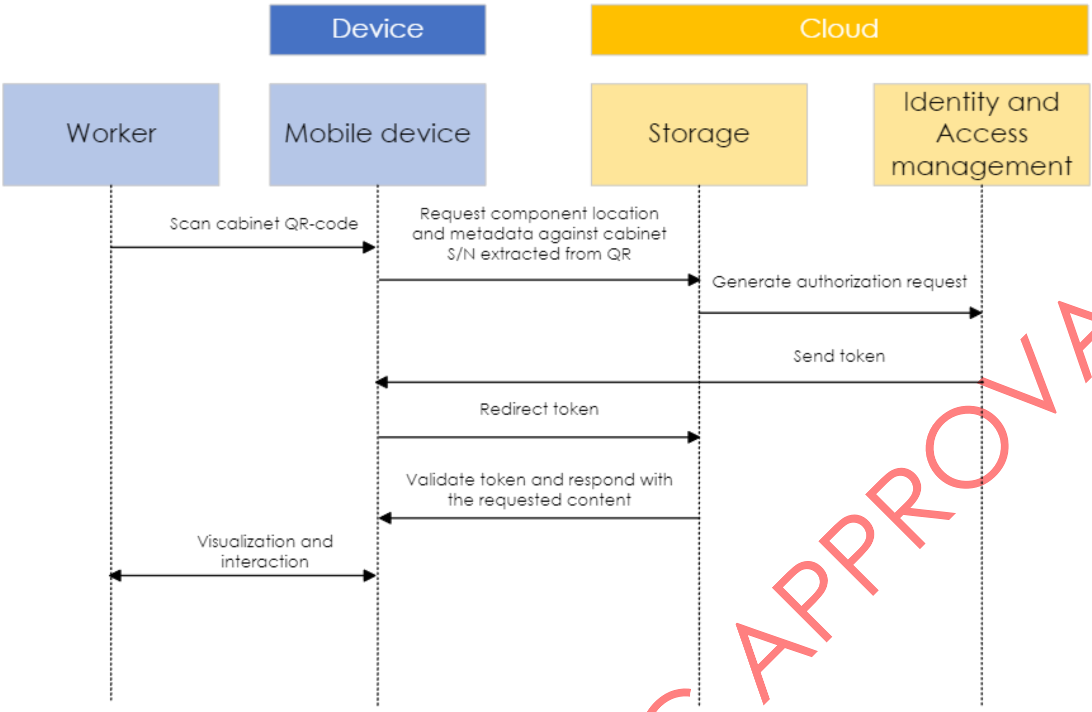


Figure 72: Sequence diagram of UC7.

3.3.2.6 Testing and validation procedures

3.3.2.6.1 Verification Testing

- The AR module and tools developed in T4.4 will be verified via the development of the AR app.
- QR codes will be used to identify the drive cabinet with the mobile app.
- Digital cabinet models usability in Smart Wiring software will be verified by the development team in a controlled environment.

3.3.2.6.2 Validation Testing

- Digital cabinet use in Smart Wiring software will be validated on a production line with production personnel by running multiple pilots.
- A survey will be conducted to evaluate the user experience and whether the new tools have improved productivity, quality, and overall working conditions.
- Service personnel will validate the use of the digital cabinet models in the AR application in a controlled environment. Learnings are used to evaluate in which area the technology would benefit the most.

3.3.2.7 Execution timeline

The execution timeline of UC7, divided into multiple phases, is detailed in Table 34 below.

Table 34: Execution timeline of UC7.

Phase	Estimated start date	Estimated end date	Notes
Trial set-up and equipment procurement	M13	M21	- Server setup - AR tools (IDE / SDK)
Initial implementation and testing	M16	M20	- First complete digital model
Intermediate implementation, testing and validation	M21	M27	- Initial version of AR app. (delayed) - Creating another digital cabinet model - Digital cabinet and Smart Wiring verification by dev. team - Digital cabinet and Smart Wiring validation by piloting in production
Final implementation and validation	M27	M34	- Digital cabinet and Smart Wiring validation by running pilots in production - Creating more digital models - AR app. verification testing - AR app. validation by service

3.3.2.8 Intermediate results

Two digital cabinet models have been developed to the point where piloting with them in production is possible. Two more are being developed.

The use of digital cabinet in Smart Wiring software and the required functionalities have been verified by the development team. Validation testing on the production line has also been done. Two pilots on production line were conducted where the cabinet construction and assembly steps were visualised for the production personnel. The feedback from users has been very positive and the visual models are a great addition to a simple list representing the required wiring. At least two more digital cabinets are being developed. The challenge is to make the digital cabinet creation light enough to be able to use them with engineered-to-order products as well.

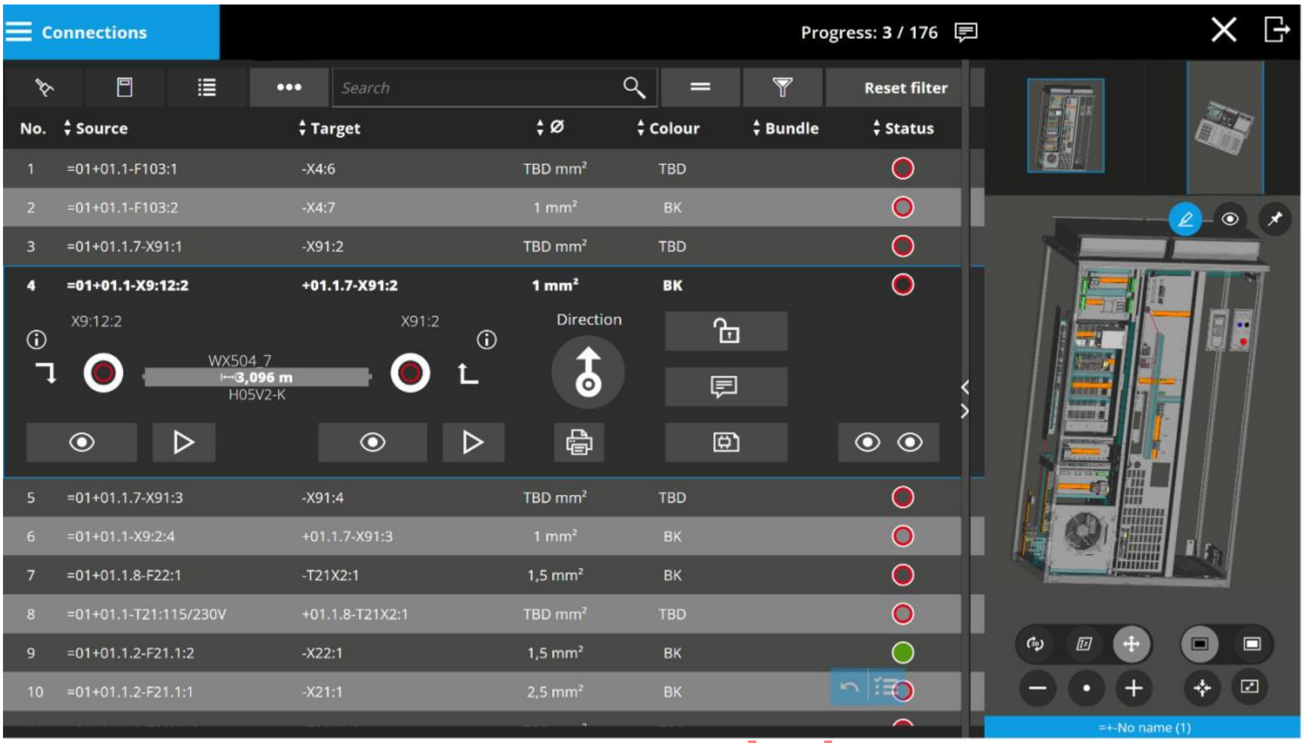


Figure 73: Example photo from Smart Wiring software with Digital Cabinet.

Digital cabinet model importing to AR application has been tested with Unity[4] and Vuforia Studio[5]. The import was successful with both IDEs, however Unity provided more powerful tools for programmatically manipulating the individual mechanical and electrical components of the model, e.g., highlighting or hiding a specific component or part.

Different potential use cases (training, sales, manufacturing and/or maintenance) for AR were analysed and the maintenance-related one was selected. In maintenance or service tasks the potential for AR is huge since the jobs on hand are diverse. By giving exact, unique product-related, and appropriately timed information the service person can be greatly helped. Maintenance use case for AR application has been described in more detail for development team and solution for service personnel validation is being developed. Other functions will follow the solution closely to identify potential for their purposes as well.

3.3.3UC8 – Digital powertrain and condition monitoring

In the context of this use case, the term powertrain is used to describe the equipment involved in transforming energy provided by a power source into useful work done by some machine. In industrial applications, such equipment typically includes an AC motor and a variable speed drive responsible for its control. Aside from direct process control, data gathered in such powertrain applications is also used for higher-level supervisory tasks and condition monitoring. The goal in this use case is to leverage IoT devices, 4G

[4] Augmented Reality Development Software – Unity : https://unity.com/unity/features/ar
[5] Vuforia Studio Augmented Reality for Industrial Enterprise – PTC: https://www.ptc.com/en/products/vuforia/vuforia-studio

telecommunications and cloud platforms to utilize novel ideas in the area of data engineering, analytics and condition monitoring.

3.3.3.1 Trial site description

The use case will take place at ABB's high-power drives laboratory located in Helsinki. ABB has two example factories, A and B. Each factory has three powertrains, from which sensor data is gathered to a gateway device. The goal is to create a holistic view of the condition and status of each powertrain, especially the drive unit (device that controls the motor) and the motor itself. Instead of using a traditional data-siloed site-specific approach, a decentralized and federated approach is taken, leveraging the IoT-NGIN paradigm and technologies.

Goal 1: A condition monitoring application needs to be able to access the sensor data gathered from powertrains (located at any site) in order to produce analytics results that can be used to monitor the condition of the devices.

- Only the device/site owner (ABB) should have access to this data.
- The solution should be scalable to support the addition of new devices and sites.
- The application needs to be able to access the data sources for each powertrain.
    - A systematic approach is needed to crawl for and access data endpoints programmatically.

One of the available hardware setups in the powertrain lab is depicted in Figure 74.

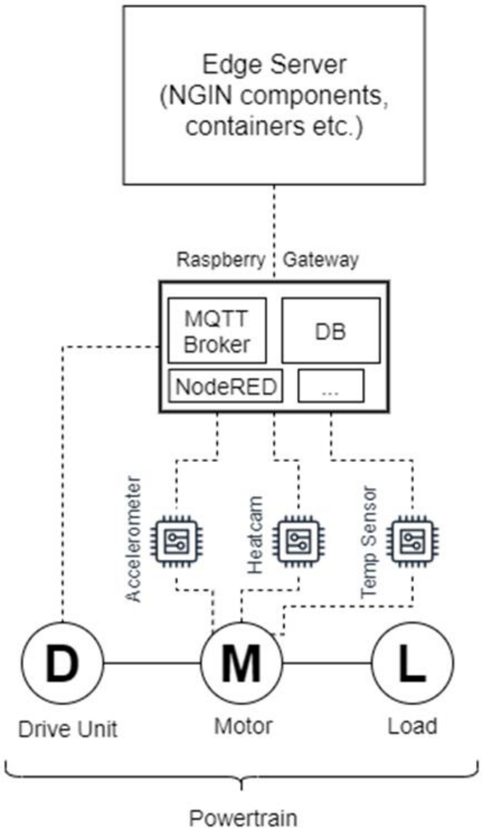


Figure 74: A laboratory setup for a powertrain.

3.3.3.2 Required equipment

Table 35 shows a detailed description of the required equipment for UC8, along with its type and status (whether it is available or to be procured).

Table 35: Detailed description of the required equipment for UC8.

Equipment	Description / Specifications	Type	Status
Edge server (Windows)	The server has the following features: • Sensor data collection • RTDB • Twinbase • GS1	Server	Available
Edge server (Linux)	The following features are available on the server: • Sensor data collection • Node RED • Device Indexing • Access Control	Server	Available

Raspberry Pi gateway	Used to collect the various sensor data gathered from a powertrain ensemble.	Gateway	Available
Drive Units	Drive units used for motor control. Include IoT-panels which can be used to send data via 4G/5G.	End device, sensor	Available
Motors	The powertrains in the lab consist of 2 or 3 motors, one or two of them act as the application load, while the other is controlled by the drive device being examined.	End device	Available
Smart Sensor	A Bluetooth Low Energy (BLE) sensor that can be attached to the side of a motor. Connected to the raspberry Pi gateway.	Sensor	Available
PLC (accelerometers + temperature sensors)	A Programmable Logic Controller (PLC) device used to connect accelerometers and temperature sensors.	Sensor / HW	Available
Heat camera	A BLE sensor that generates a 2D heatmap view. Connected to the raspberry Pi gateway.	Sensor	Available

3.3.3.3 Data collection

The following dataset(s) have been identified as part of the use case:

- **Drive data:** drive operating data gathered via its IoT-Panel: speed, torque, current, voltage, etc., used for operation and condition monitoring.
- **Smart sensor data:** acceleration and temperature related KPI values.
- **PLC data:** high frequency accelerometer data and temperature data. For example, motor temperatures are collected by using PLCs.

3.3.3.4 Alignment with IoT-NGIN technologies

Table 36 below shows the IoT-NGIN technologies which are targeted for UC8. The table has been updated since D7.2. For each technology, a description of its role is provided along with any adaptation it might need to be used in the UC, deployment details and related UC requirements and KPIs.

Table 36: Alignment of UC8 with the relevant IoT-NGIN technologies.

WP4 – IoT device discovery and indexing

Description	Drive devices and/or gateway devices need to be registered and made available to other IoT-NGIN components and use case specific applications (In this case a condition monitoring application)
Adaptation and fine-tuning	The registering and availability of devices should also leverage the semantic twin concept developed in T5.5, which will be used in this use case.
Deployment	Edge server
Related requirements	**REQ_IN3_F01** – IoT-device data must be accessible to device owner from edge node to generate a holistic view **REQ_IN3_F02** – IoT-device data must be protected so that only the owner of the device has access to the data.
Related KPIs	**KPI_IN3_03** – Number of digital twins to be created using an IoT-NGIN conformant pipeline > 3
WP5 – Semantic Twins	
Description	Semantic Twins (previously referred to as Meta-Level Digital Twins) will be used to create a condition monitoring application. The application will be able to browse and process twin documents in order to access relevant data and APIs for condition monitoring purposes.
Adaptation and fine-tuning	N/A
Deployment	Local deployment / edge.
Related requirements	**REQ_IN3_F01** – IoT-device data must be accessible to device owner from edge node to generate a holistic view **REQ_IN3_F02** – IoT-device data must be protected so that only the owner of the device has access to the data. **REQ_IN3_NF01** – Support for batch data processing
Related KPIs	**KPI_IN3_03** – Number of digital twins to be created using an IoT-NGIN conformant pipeline > 3

Compared to D7.2, the list of relevant IoT-NGIN technologies has been modified. Original plan was to include ML models and related technologies from WP3 and WP4 to be used in UC8, but due to resource issues it was agreed to remove these targets.

3.3.3.5 Use case sequence diagrams

Figure 75 shows a sequence diagram of UC8, depicting the involved actors and their interactions with the relevant IoT-NGIN technologies. The following user actions have been recognized:

1. The user is present at the powertrain site and has physical access to the powertrain. It is also possible to access the system remotely if QR codes are located i.e., in the control room of the site.
2. The user is using a mobile device (e.g., cellular phone) to open an application to scan QR code attached to a system or component (or remotely). The user is able to select a specific powertrain or component if several are available. IoT Device Indexing module is used in this phase.
3. The user gives necessary data to be identified to get access to the documentation. For this IoT Access Control module is used.
4. The user has a view on the selected powertrain asset.
5. The user is able to check the content of the digital document of the selected powertrain. Information is available in the structured way according to Figure 73. The Metadata of powertrain can be examined, and different kinds of digital services are accessible via links in the documentation.
6. The user selects one of the available services, for example, temperature monitoring of an electric motor. Additional identification may be required at this point, depending on the service.
7. Instead of monitoring real-time data, the user can switch to virtual monitoring of the electric motor. The user must select a service providing access to the simulation model of the selected powertrain. Again, additional identification is potentially required at this point, depending on the service.

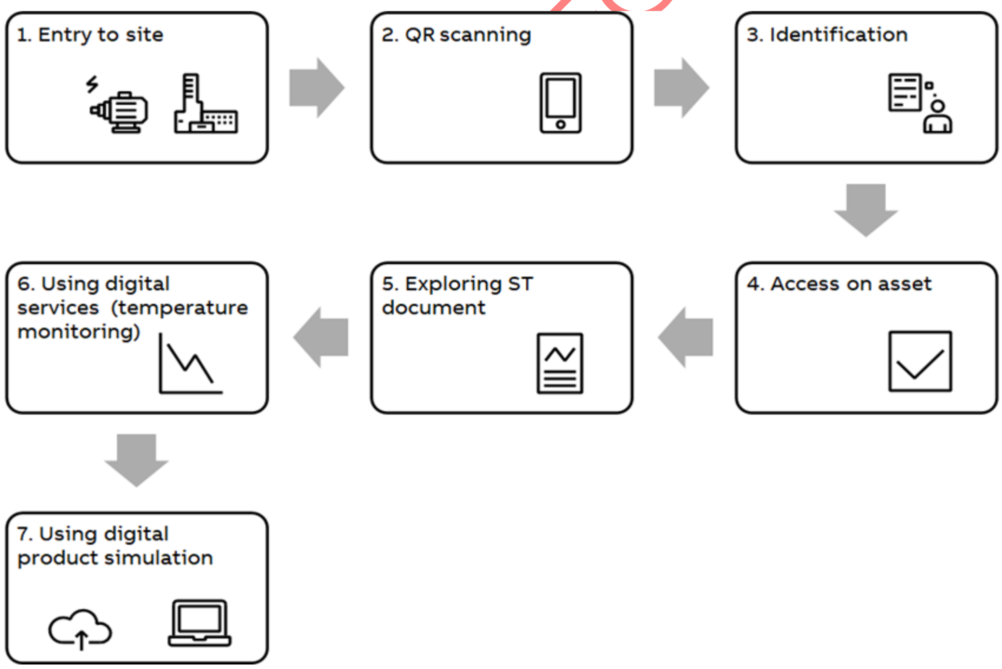


Figure 75: Flow chart of user actions using Digital Twin of a powertrain.

3.3.3.6 Testing and validation procedures

3.3.3.6.1 Technical validation of the UC

- Devices and data will be made available digitally utilizing the results of tasks 4.2 and 5.5. KPI_IN3_03 specifies that digital representations of more than 3 powertrains will be produced in this use case.

3.3.3.6.2 Validation stages

- Individual components will first be validated and tested in development environments without actual physical equipment.
- Final validation will be done in the high-power laboratory using live equipment.

3.3.3.6.3 Validation of the IoT Device Indexing & IoT Access Control

Validation will be done against the following requirements/questions:

1. Is user able to access different semantic twin documents according to given access rights?
2. Is it possible to get any data of documents if access is denied?
3. Is access control somehow visible for user? Can user be confident that access control is working?
4. Is access control working so that it does not make usage too complicated?

3.3.3.6.4 Validation of the Semantic Twin

Validation will be done against the following requirements:

1. Is user able to access different semantic twin documents?
2. Is semantic twin approach providing good framework to get data from powertrains?
3. Is semantic twin approach providing data that is useful in sense of visual presentation?
4. How easy is it to modify the structure of twin document and the ontology of the system?

3.3.3.7 Execution timeline

The execution timeline of UC8, divided into multiple phases, is detailed in Table 37.

Table 37: Execution timeline of UC8.

Phase	Estimated start date	Estimated end date	Notes
Trial set-up and equipment procurement	M10	M16	- IoT-panel FW modifications - Procuring hardware and data handling implementations on gateway device
Initial implementation and validation	M16	M20	- Procure edge server - Complete semantic twin document of a powertrain and initial deployment of a semantic twin pipeline/architecture.
Intermediate implementation and validation	M21	M27	- Integration of semantic twin with other IoT-NGIN components.

Final implementation and validation	M27	M34	- Integration of semantic twin with other IoT-NGIN components
Final implementation and validation	M30	M34	- Integration of IoT Device Indexing and Access Control module

Compared to original timeline presented in D7.2, the following changes have been noted:

Original edge server has been implemented at early stage (before M20). Due to some new requirements, a new edge server solution was implemented until M27.

Due to lack of resources, the implementation and validation of IoT Device Indexing and Access Control modules will happen during M30 - M34.

3.3.3.8 Intermediate results

At this stage of the use case, the following results have been obtained:

- Powertrains are measured by modified IoT panels and several other sensors so that data can be gathered flexibly and utilized by IoT-NGIN components.
- A raspberry PI gateway device has been prepared, which can process and then forward data to IoT-NGIN components, avoiding possible integration problems down the line.
- Edge server solution has been implemented to fetch and manage data, and to run the required software.
- An initial twin document has been prepared describing data endpoints of a powertrain ensemble using the W3C WoT model.
- A condition monitoring SW has been developed to collect and visualize data from powertrains.

In the following, an example is given how semantic twin and other solutions are implemented in use case. One typical use case is that the user wants to check the status of the electric motor in operation. Temperature of the motor is a good indicator about the status of the motor. Node-RED view of temperature behaviors of selected motor is shown in Figure 76. The user can check real-time temperature of the motor, but a histogram is also available. The temperature data can be used to indicate the following things:

1. Data can be used to check if there are any environmental changes visible
2. Need for maintenance actions can be checked, e.g., if predefined temperature limits have been crossed
3. Comparison between different installation can be made
4. Malfunction of sensors can be indicated

Generally, digital services like temperature monitoring can be seen as additional value for different stakeholders of powertrain. Powertrain operators, maintenance service, manufacturers and so on can benefit from this kind of service.

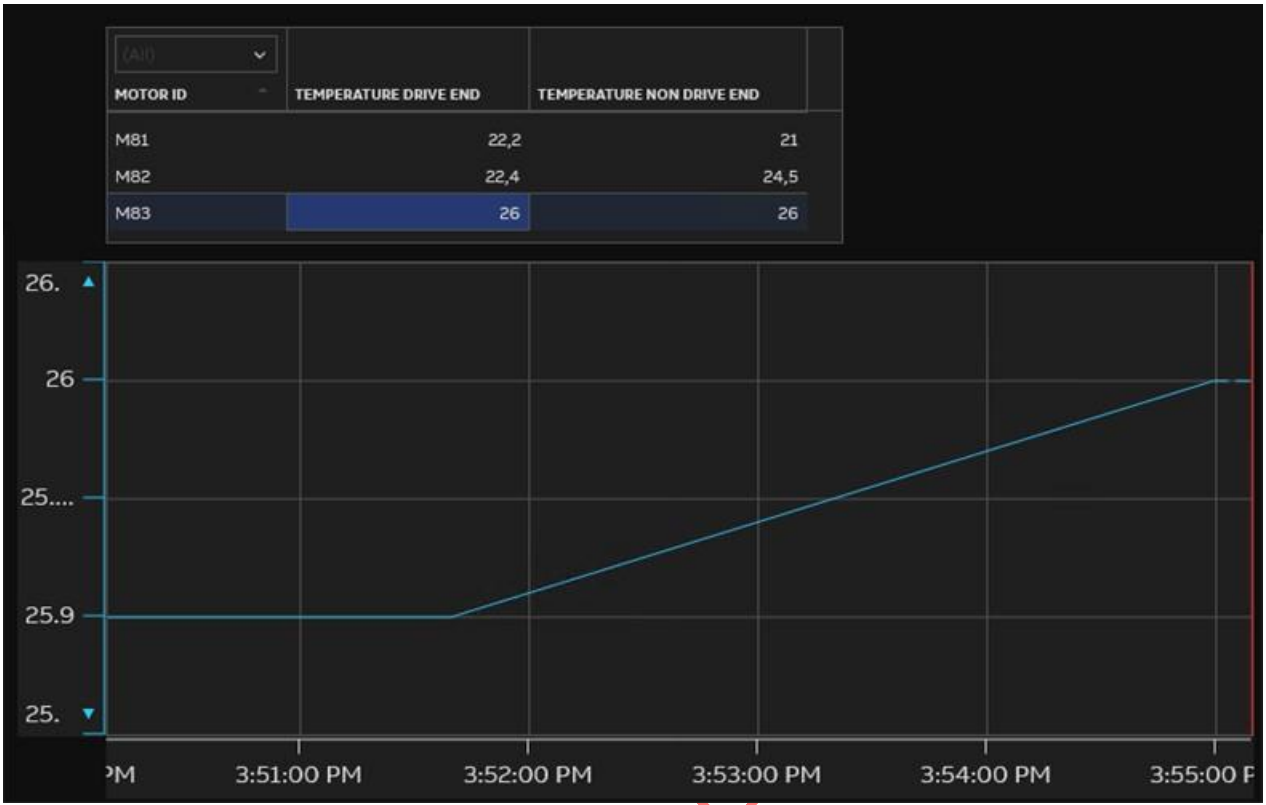


Figure 76: View on temperature of electric motor.

At this point it is obvious that semantic twin approach is a good framework for UC8. In the future, the following aspect should be considered:

- How to modify document content easily as there are lot of variations in powertrain setups? Manual work should be avoided, and user-friendly interface should be available for operators.
- What is the automized procedure to add new data to twin documents?

Validation results of IoT Device Indexing and Access Control are not yet available for UC8 and those will be given in D7.4.

3.4 Smart Energy Grid Monitoring / Control Living Lab

3.4.1 UC9 – Move from reacting to acting in smart grid monitoring and control

The Smart Energy LL deals with the integration of IoT devices, and related services, within the electricity distribution network, assessing both the role of demand response and the impact and potential of electric mobility.

The Smart Energy LL tests different types of services, related to cybersecurity, Machine Learning as a Service, Augmented Reality and some user interface services.

Electricity distribution is a complex system that is closely linked with the lives of end users and its management, especially in a period of energy transition towards renewable sources, requires steps towards real-time monitoring of parameters and the integration of innovative services, and the IoT plays a decisive role in this area.

3.4.1.1 Trial site description or update

The Smart Energy Living Lab is located in Terni, central Italy, and is conducted on the electricity distribution network, which is operated by ASM Terni. ASM is a multi-utility and plays the role of DSO. The purpose of this Living Lab is to address some issues related to the electric distribution network, namely the diffusion of electric vehicles (EVs) and demand response mechanisms. The electric grid due to the massive diffusion of renewable sources has become more prone to imbalances in terms of voltage (surges and under voltages), current (overloads), and power (reverse power flow peak phenomena). EVs can have significant power impacts if not properly managed, but they can also provide ancillary services to the grid by shifting load during peak generation times. Demand Response (DR) is a mechanism to enable active user participation by providing some ancillary services to the grid.

In Terni's smart grid, there are several sensors, such as smart meters, phasor units, and power quality analyzers. The data are managed by a SCADA system, which collects all the data and enables efficient sharing with partners.

Compared with previous deliverables, in the Smart Energy LL, the updates were in terms of development in relation to technology alignment with technical partners, so some sensors were changed, and some data were added, and there are also intermediate results of service integration.

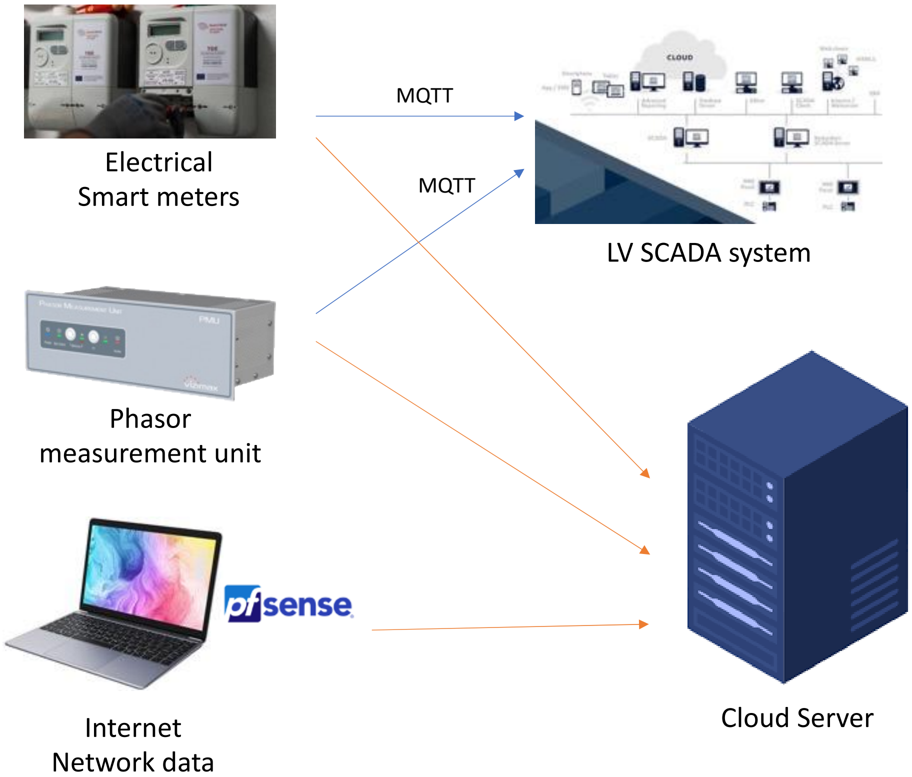


Figure 77: A graphic description of the trial site of UC9.

3.4.1.2 Used equipment

Table 38 below shows a detailed description of the required equipment for UC9, along with its type and status (whether it is available or still to be procured).

Table 38: A detailed description of the required equipment for UC9.

Equipment	Description / Specifications	Type	Status
A portion of the electrical distribution grid	The portion of the grid used for the test service of Atos is an MV grid with 14 buses.	Infrastructure	Available
ASM server	It is a SCADA system made by Wonderware, able to monitor in real-time hundreds of sensors and store historical data.	Server	Available
Smart Meter	43 electrical smart meters were placed in the condominium, able to monitor active and reactive power and voltage	Sensor	Available

PMU	1 phasor measurement unit, able to monitor active and reactive power, voltage, phase angle, harmonic content, etc	Sensor	Available

The equipment remains the same as in D7.2, but the quantities have been changed, as data from some smart meters and a PMU are not available and have therefore been removed.

With respect to D7.2, in working towards integration with the technical services, it was appropriately defined which sensors to use for the different services, their location and their role.

3.4.1.3 Data collection

The following datasets have been identified as part of the use case:

- **Smart Meter eXtension (SMX) dataset:** Measurements of voltage, currents, and power derived by ASM energy units, collected through an MQTT protocol via public IP and/or LAN/VPN connections. To ensure secure access to the dataset, credentials will be required.
- **Phaser Measurement Units (PMU) dataset:** Measurements of voltage, currents, and power derived by ASM energy units, collected through HTTP protocol only via LAN/VPN connections, which provide built-in security access features.
- **Internet network dataset:** Few years' worth of data is available on the activities of the open-source firewall pfSense, used to manage the Internet access of a group of sensors located at an apartment building. The Internet data are related to pfSense reports and a Suricata dataset.

Compared to D7.2, the Internet data set was added to test some cybersecurity services.

3.4.1.4 Alignment with IoT-NGIN technologies update

Table 39: Technology alignment for UC9.

WP3 – MLaaS framework	
Description	The MLaaS framework will be used to optimize the operation of the digital twin and choose the most appropriate flexibility tool for the electrical grid. ML as a Service Framework includes data acquisition, data pre-processing, AI modelling (at the edge), AI model deployment, integration and model operation, and model sharing.
Adaptation and fine-tuning	The technology will be developed using specific machine learning tools for optimizing electricity grids.

Deployment	The service will be carried out on the cloud and will be based on the data that is communicated by ASM and the digital twin implemented in another service. It will act on the parameters of the digital twin to optimize them according to the needs of the DSO. The model was created by ATOS and for other nodes, they will only use the historical data that has been provided by ASM.
Related requirements	**REQ_SE1_F02** – High-tech power sensors should be useful to elaborate on new strategies, to improve the power quality in a secure way. Smart Meter should help this process
Related KPIs	**KPI_UC9_4** – Reduce the probability of Smart Grid failure due to voltage instability by at least = 25 % reduction in comparison with the daily average value **KPI_UC9_5** – Increase urban Electrical Vehicles charging efficiency = 20 % in comparison with a daily average value
WP3 – Deep Learning, Reinforcement Learning & Transfer Learning	
Description	Deep learning and reinforcement learning techniques are used to enhance training processes with adaptive self-learning. To anticipate when there could be surges, under voltages, or other defects based on demand and generation, ML approaches will be employed for Digital Twin's generation and consumption forecasting.
Adaptation and fine-tuning	Based on the extensive historical data, a forecast of generation or consumption at network nodes will be assessed.
Deployment	A model of the network in MV was made first with OpenDSS and then with pandapower, to make the power flow of the network. Atos is adding a machine learning component to optimize load shift, through demand response and better manage the network. Constraints will be added to demand response to limit the things that can be asked of users. Based on historical data from electrical sensors, this service will be implemented in the cloud.
Related requirements	**REQ_SE1_F02** – High-tech power sensors should be useful to elaborate on new strategies, in order to improve the power quality in a secure way. Smart Meter should help this process
Related KPIs	**KPI_UC9_4** – Reduce the probability of Smart Grid failure due to voltage instability by at least = 25 % reduction in comparison with a daily average value
WP4 – Device indexing	

Description	Device indexing is used to provide real-time updates on IoT devices, as well as to monitor inactivity and the durability of historical data.
Adaptation and fine-tuning	It is based on the FIWARE Orion COntext broker.
Deployment	Access and log sensor data and then it is communicated with the dashboard. ASM provided access to available sensors in the condominium. In addition, ASM has granted SYN access to the firewall and computer which are available in the apartment building, so that we can do the ssh commands.
Related requirements	N/A
Related KPIs	N/A
WP4 – Device access control	
Description	Device access control is employed to control who has access to the digital twin (once it works it will be connected with Self Sovereign Indexing, which will work as a plugin).
Adaptation and fine-tuning	There are 3 plug-ins: SSI, proximity, and Keycloak. Access policies are determined.
Deployment	Device Access Control is like a proxy that protects access to MQTT, it is a flexible gateway. ASM provided access to available sensors in the condominium. In addition, ASM has granted SYN access to the firewall and computer which are available in the apartment building, so that we can do the ssh commands.
Related requirements	N/A
Related KPIs	N/A
WP5 – Malicious Attack Detector (MAD) and GAN-based dataset generation	
Description	Malicious attack detection at the network level applied to the data management platform of IoT-NGIN. A Malicious Attack Detector (MAD) is developed with the primary goal of defending IoT systems against assaults. The Generative Adversarial Network (GAN)-based algorithm used by MAD first learns the datasets for poisoning before seeing hostile activity. MAD will be created and implemented into the system to identify malicious nodes to safeguard the security of IoT networks. Based on a two-player game-theoretic scenario, GAN models are employed for unsupervised learning. A GAN model's goal is to learn the distribution and patterns

	of the training data so that it can produce new data that has the same properties as the training data.
Adaptation and fine-tuning	Systems will be used to detect attacks and protect the platform from a cybersecurity perspective.
Deployment	Cybersecurity services produced within the data management platform will be tested. ASM has shared with SYN the dataset of meerkats and Pfsense. Sharing data in real-time is desired if it is possible, otherwise, the data must be sent once a month alternatively.
Related requirements	**REQ_SE1_NF02** – Secure communication of sensitive data related to the infrastructure should be provided
Related KPIs	N/A
WP5 – Privacy-preserving Self-Sovereign Identities (SSIs)	
Description	Selfie Sovereign Identities (SSI) aims to provide users authority over their identities so that identities may be created independently of another source. This greatly enhances user privacy by enabling users to create and use identities without any outside entity watching their behaviour across numerous services. SSIs will integrate techniques to limit access to the grid's digital twin and allow ubiquitous security.
Adaptation and fine-tuning	No adaptation need was identified at this point.
Deployment	Collaboration with AALTO will be done when the device access control service is ready.
Related requirements	**REQ_SE1_NF02** – Secure communication of sensitive data related to the infrastructure should be provided
Related KPIs	**KPI_UC9_3** – Event detection time using the digital twin concept < 50 ms
WP5 – IoT vulnerability crawler	
Description	This component will identify potential vulnerabilities in ASM sensors
Adaptation and fine-tuning	No adaptation needs to be identified at this point.
Deployment	Edge/Cloud ASM shared the list of IPs for all the sensors in the condominium.

Related requirements	**REQ_SE1_NF02** – Secure communication of sensitive data related to the infrastructure should be provided
Related KPIs	**KPI_UC9_3** – Event detection time using the digital twin concept < 50 ms
WP6 – Visualization dashboard	
Description	The MV network's parameters (estimated and optimized), as well as the data from smart meters (only real data). The dashboard, however, will be used to gather and display all the services provided. The goal is to share the results of the forecasting service in the dashboard in real-time. For example, after publishing the results on an MQTT Broker, we could read them from the dashboard and plot them.
Adaptation and fine-tuning	We will use it for the UC9 and part of the UC10, we will show the PMU data (real and expected)
Deployment	This dashboard will be used to show real-time data and data resulting from forecasting and optimization services. This dashboard will need to connect with MQTT and ASM SQL. List of required sensors have been determined to be displayed, divided between actual, forecasted, and optimized. A draft version of the dashboard is ready which needs to be extended for all sensors.
Related requirements	N/A
Related KPIs	**KPI_UC9_2** - Information exchanged by devices > 100000 measurements/minute.

3.4.1.5 Use case sequence diagram(s)

The sequence diagram in the following figure shows the current situation of the connection of data sources and services.

Compared with D7.2, several changes are present, as the definition of services has made it possible to clarify better the integration between data sources, services, and the display of results.

The services related to WP2 were removed, as these will be tested elsewhere and are no longer in this LL. Several cybersecurity services (such as the Malicious attack detector, the GAN-based IoT attack dataset generator, and the IoT vulnerability Crawler) and some services related to the digital twin (IoT device access control and IoT device indexing) were included.

Some of the most important results will be displayed in a dashboard, which will show historical values of actual data, predicted data, and optimized data.

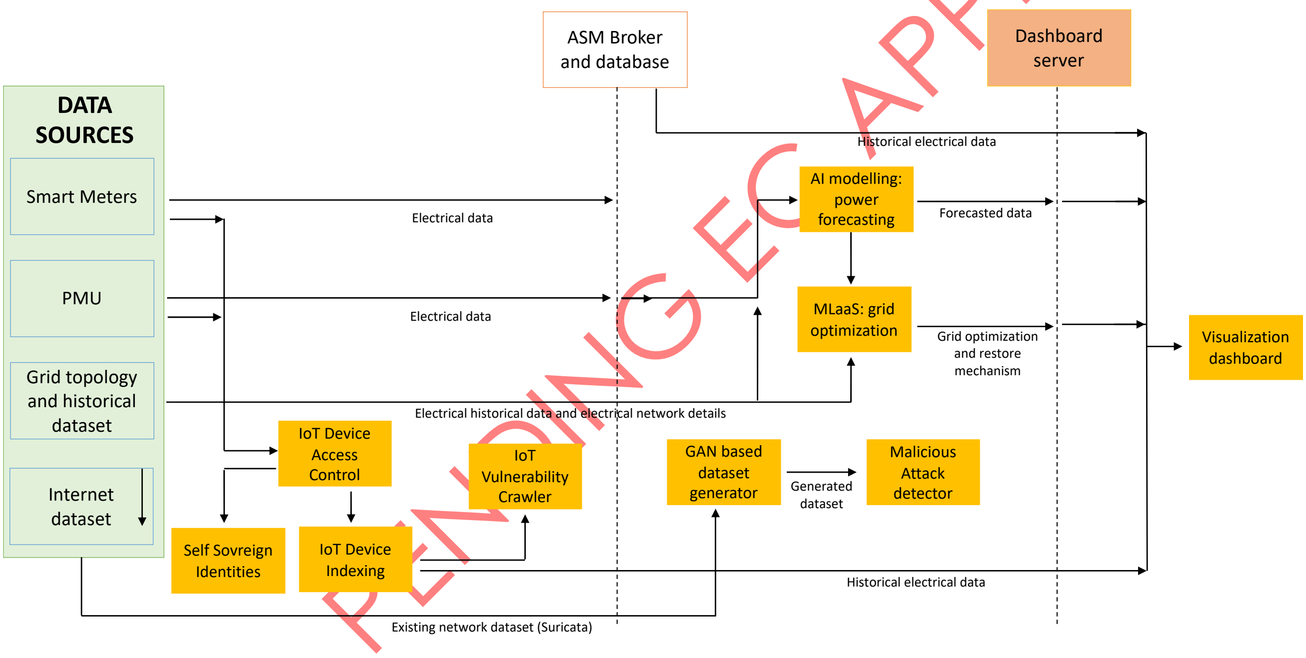


Figure 78: Sequence diagram of UC9.

3.4.1.6 Testing Scenarios

For validating the IoT-NGIN technology developed in the Smart Energy area three test scenarios have been defined and are being implemented in the LL with the use of the IoT-NGIN tools. The test scenarios are described in the following subsections.

3.4.1.6.1 Energy generation and consumption forecasting

Table 40: UC9 Test 1 "Energy generation and consumption forecasting".

Test 1: Energy generation and consumption forecasting

Objective	Determine the power flow that will pass through the main feeder in the next 24 hours, so as to have a better knowledge of the network parameters and to be able to guide operators in their decision-making.
Components	• ASM Server • PMU • MLaaS framework • Visualization dashboard
Features to be tested	• ability of the system to provide forecasts of the active and reactive power passing through the primary cabin • possibility to visualize results via a dashboard
Requirements addressed	REQ_SE1_F02
Steps	1) Realisation of the forecasting model based on ML 2) Model training on ASM dataset 3) Analysis of results and improvement of the model 4) Visualization of the results in a dashboard
KPIs	KPI_UC9_4, KPI_UC9_5, KPI_UC9_2

3.4.1.6.2 Grid operation optimization

Table 41: UC9 Test 2 "Grid operation optimization".

Test 2: Grid operation optimization	
Objective	Optimize the power flow of the utilized portion of the grid by leveraging energy flexibility resources through demand response mechanisms. This reduces the impact on the grid in terms of overloads, undervoltages, and overvoltages.
Components	• ASM Server • PMU • MLaaS framework • Deep Learning, Reinforcement Learning & Transfer Learning • Visualization dashboard
Features to be tested	• effectiveness of the ML-based network optimization service. • possibility to compare the results of the optimization service and non-optimized data via a dashboard
Requirements addressed	REQ_SE1_F02
Steps	1. Realization of the electrical simulator model using pandapower 2. Realization of the optimization model based on ML 3. Model training 4. Analysis of results and improvement of the model 5. Visualization of the results in a dashboard

	In the test scenario, different DR penetration scenarios will be considered, allowing for increased use of this technology for the provision of ancillary services to the electricity distribution network. DR mechanisms are used within the electricity grid optimization algorithm, to limit the phenomena of Undervoltage, overvoltage, overload, to increase the grid's self-consumption, and to limit power losses. The following three scenarios will be tested, considering the increasing spread of the DR program. • it is considered that participation in demand response mechanisms is reduced: 10% of the consumer's load can be shifted over time by means of demand-side management mechanisms • it is considered that participation in demand response mechanisms is moderated: 20% of the consumer's load can be shifted over time by means of demand-side management mechanisms • it is considered that participation in demand response mechanisms is high: 30% of the consumer's load can be shifted over time by means of demand-side management mechanisms
KPIs	KPI_UC9_4, KPI_UC9_5, KPI_UC9_2

3.4.1.6.3 Demand Response integration

Table 42: UC9 Test 3 "Demand Response integration".

Test 3: Demand Response integration	
Objective	Enable the integration of sensors within the digital network system and securely guarantee the transit of information. Assess network weaknesses and take the necessary countermeasures to ensure user privacy and the veracity of information.
Components	• ASM servers • end users' smart meters • ASM firewall dataset • Device indexing • Device Access Control • Malicious Attack Detector (MAD) and GAN-based dataset generation • Privacy-preserving Self-Sovereign Identities (SSIs) • IoT vulnerability crawler • Visualization dashboard
Features to be tested	• integration of sensors within the digital network model • vulnerability of measurement devices • vulnerability of the private Internet network • visualization of historical data of measuring devices
Requirements addressed	REQ_SE1_F02

Steps	As far as the analysis to be carried out on the sensors is concerned, proceed as follows: 1) Install the device access components in a server 2) Populate the server with sensor data 3) Integrate sensors into device indexing and device access control 4) Insert the plug-in of the Privacy-preserving Self-Supervised Identities into the IoT device access control 5) Vulnerability analysis of IoT sensors 6) Visualisation of historical data collected in IoT device indexing With regard to analyzing the cybersecurity of the private internet, the following steps are taken: 1) Collecting internet flow data via Suricata and the pfSense firewall, 2) Integration of MAD services and generation of GAN-based datasets 3) Network vulnerability analysis
KPIs	KPI_UC9_2, KPI_UC9_3

3.4.1.7 Execution timeline

UC9 is currently at an intermediate stage, where services have been well defined and integrated, although results are still limited.

As a first activity, the integration of all technical services tested in UC will be completed (by M31) and then the Quality-of-Service integration will be continuously increased in the following months (M31-M33).

Some services will operate continuously, acquiring real-time data, running dashboards, and performing real-time analyses, so their management will be continued until the end of the project. Final results are expected to be achieved by M33, and then extended and generalized in the following months. Dissemination and business exploitation activities will take place soon after the final results are obtained.

UC 9	apr-23	may-23	jun-23	jul-23	aug-23	sep-23
Evaluation Framework	**31**	**32**	**33**	**34**	**35**	**36**
1. Strategy and general context						
KPI impact calculated						
Evaluation of impact achieved						
2. Digital technologies for the Living Lab						
Complete the integration of technical services						
Refine technical services based on implementation feedback						
Continuous updating of services and server support						
Obtain the final results						
Extending final results on a large scale						
Business exploitation plan						
Dissemination of the results						

Figure 79: UC9 Execution timeline.

3.4.1.8 Intermediate results

In UC9, data from two types of electrical sensors are used:

- smart meters, which measure the active and reactive power of domestic end-users with a granularity of 5 seconds.
- Phasor measurement units, measure the current for each phase, the voltage for each phase, the frequency, and the harmonic content in the MV feeder in the primary cabin.

In addition, internet data is available from a subsection of ASM's internet network, i.e., the one exclusively dedicated to the connection of smart meter sensors. This network uses the pfSense firewall. In UC9, the technical services related to machine learning, cybersecurity services, as well as visualization services developed within IoT-NGIN are applied to support *Energy generation and consumption forecasting*, *Grid operation optimization*, and *Demand Response integration* scenarios.

The intermediate results of each of the three categories are presented below.

3.4.1.8.1 Energy generation and consumption forecasting

The services based on machine learning are AI modelling for power forecasting and MLaaS for grid optimization.

As far as forecasting is concerned, the prediction of the voltage trend and the active and reactive power of the PMU, located near a primary substation, which feeds the portion of the grid concerned by the analysis, was carried out. Forecasting was carried out using data from the last months, by dynamically training a RNN model that consists of several GRU layers, with the help of the MLaaS Online Learning service, as reported in D3.3. It was chosen to have predicted data with a time horizon of 24 hours and a resolution of one hour, whose forecasting is based on the last 36 hours of data averaged every hour, resulting in an input tensor of 36 values.

D3.3 provides details on the online learning implementation and experimentation settings, as well as on the forecasting results obtained for the UC9 power generation, which are shown in Figure 80.

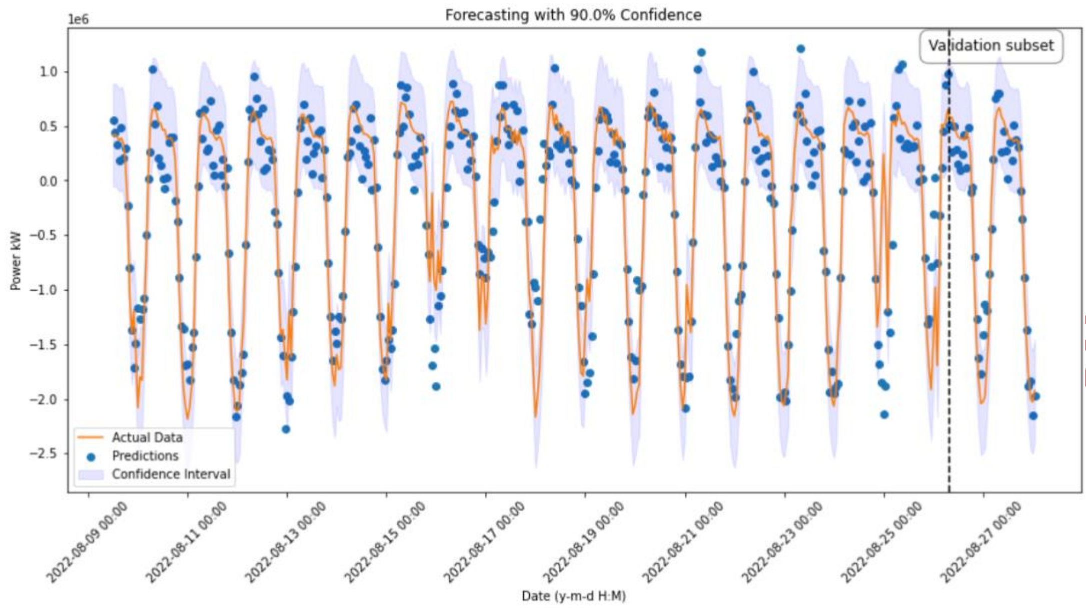


Figure 80: Training results for power generation forecasting of UC9.

Moreover, Explainable AI (XAI) techniques have been adopted in the OL service and reported in D3.4 to explain the influence of the input features (i.e., previously measured power generation metrics) on the forecast, resulting in the insight that the more recent values have a stronger influence in the prediction that the older ones.

Additionally, as reported in D3.4, the sustained suitability of the forecasting model to the UC9 power generation data along the time has been measured with the MLaaS ML training monitoring system, resulting in the no detection of data drift during the online training of this model. This fact gives confidence in the predictive power of the model with time-evolving datasets. Similarly, as reported in D3.4, the same monitoring system has been used to inspect the time-sustained model performance gain as new incoming datasets have been used to further train the forecasting model.

3.4.1.8.2 Grid operation optimization

A power grid solver, connected to a machine learning model, is used for grid optimization.

The power grid solver was written both through the open-source software OpenDSS[6], which runs in the Windows operating system and by interfacing it with Python via the py-dss-interface package.

Although the results were good, it was decided to use another software, which operated directly in the Python environment, without needing the Windows operating system. The choice was therefore made to use pandapower[7], a python module used for solving electrical networks. In the case study used, the two tools provide the same results and are equivalent.

[6] Open Distribution System Simulator, available online at: https://sourceforge.net/projects/electricdss/
[7] Available online at: http://www.pandapower.org

The electricity network solver acquires as input the historical day-by-day consumption and generation data of clusters of users (13 clusters of industrial users, 13 clusters of domestic users, and 13 clusters of photovoltaic generation plants) and outputs the results on the parameters to be optimized: power losses, self-consumption rate, self-sufficiency rate, the energy of reverse power flow.

A Reinforcement Learning (RL) based optimization model has been designed and implemented, for learning the optimal daily power demand profile for domestic and industrial users of the electric network. Details of the design and implementation have been reported in D3.4. Details on the application of this RL-based optimization model to this Smart Energy UC9 for grid optimization have been also provided in D3.4, including the description of the environment, states, actions, and rewards.

In the following, we summarise the most relevant use case specific aspects of the optimization model setup and report on the first optimization results.

The optimization model interacts with a simulated version of the electric grid, through the pandapower-based simulator aforementioned. The model shifts the power demand profile of the domestic and industrial clusters by applying actions selected from a given action set, injecting those profiles into the simulator, and receiving an updated environment observation, as a result of simulating the electric network with the new power demand. As part of these observations, two relevant metrics are reported, the SSR (self-sufficiency ratio) and SCR (self-consumption ratio). The optimization model operates on the power demand, according to certain criteria, modifying the trend over time of the electrical power demanded by the user clusters and indicating what the optimized power flow should be, to be achieved through demand response mechanisms.

The criteria and constraints used by the model are as follows:

1. The power demand can be only shifted to a close time slot within a maximum of two hours,
2. The power demand cannot be negative,
3. The total daily power demand is constant for each cluster.

The power demand profile describes the daily demand, with a 15-minute interval, resulting in a 96 values vector for a full day.

The distribution of the demand (i.e., its profile) for the different clusters determines the environment states. In order to simplify the state space, the average power demand distribution for domestic and industrial loads is considered, as explained in D3.4).

The action set is built as the product of eight specific actions subsets, namely:

- The selection of the domestic cluster, among 4 select cluster demand profiles.
- The selection of the industrial cluster, among 4 select cluster demand profiles.
- The selection of the initial time for the demand shift for the domestic cluster, discretizing the day into hours.
- The selection of the initial time for the demand shift for the industrial cluster, discretizing the day into hours.
- The selection of the final time for the domestic demand shift,
- The selection of the final time for the industrial demand shift,
- The amount of demand shift (positive or negative) in percentage for the domestic cluster, with values in set +/-[2, 5]%.

- The amount of demand shift (positive or negative) in percentage for the industrial cluster, with values in set +/-[2, 5]%.
-

In Table 43 we list the different actions contributing to the whole action set.

Table 43: Action subset.

Action subset	Type	Length
Domestic demand cluster selection	Discrete	4
Industrial demand cluster selection	Discrete	4
Domestic initial time slot	Discrete	24
Industrial initial time slot	Discrete	24
Final domestic time slot	Discrete	2
Final industrial time slot	Discrete	2
Demand shift for domestic cluster	Discrete	2
Industrial shift for domestic cluster	Discrete	2

The reward is computed as the normalized sum of SSR and SCR: R=0.5(SSR+SCR)

3.4.1.8.3 Preliminary experiments

Initial experiments for training the optimization model have been conducted. In Table 44 we describe the hyperparameters of the model agent.

Table 44: Hyperparameters of the model agent.

Parameter	Value
RL Algorithm	Proximal Policy Optimization
Optimizer	ADAM
Batch size	1
Learning rate	0.001
Multi step	10
Exploration rate	10%
Reward discount	0.99

Experiments have been initially conducted with a small number of episodes, in order to reduce the experiment time so that the correctness of the implementation can be evaluated and draw some initial results on the efficiency of the optimization training. Parameters describing the experiments are listed in Table 45.

Table 45: Experiment parameters.

Parameters	Value
Number of episodes	5
Steps per episode	1500
Action subset size	8
State subset size	2

Table 46 shows some details about experiment timing, including the average step duration, the average episode duration, and the total experiment duration.

Table 46: Experimental timing.

Parameter	Value
Average step duration	0.26 seg
Episode duration	394 min (6h33min)
Total experiment duration	1970 min (32h50min)

During this preliminary experiment the optimization model was trained, getting an increasing reward over the episodic evolution, as shown in Figure 81 where the reward is displayed for the three different episodes.

Figure 81: Reward evolution over the training process.

It can be observed the reward grows over the episode training process, while the optimization model computing the optimal policy that determines the best action to take for a given state.

In Figure 82, we render the initial (on the left) and the optimized (on the right) demand profiles for one domestic cluster, after one episodic training.

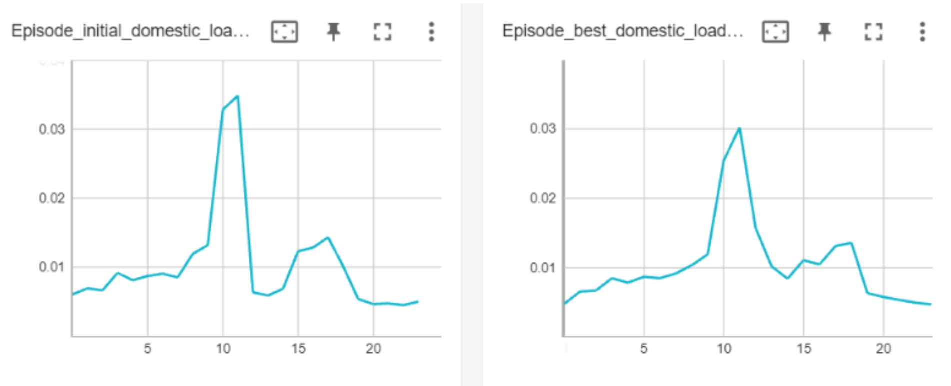


Figure 82: Domestic demand profile before and after the training process.

In Figure 83, we render the initial (on the left) and the optimized (on the right) demand profiles for one industrial cluster, after one episodic training.

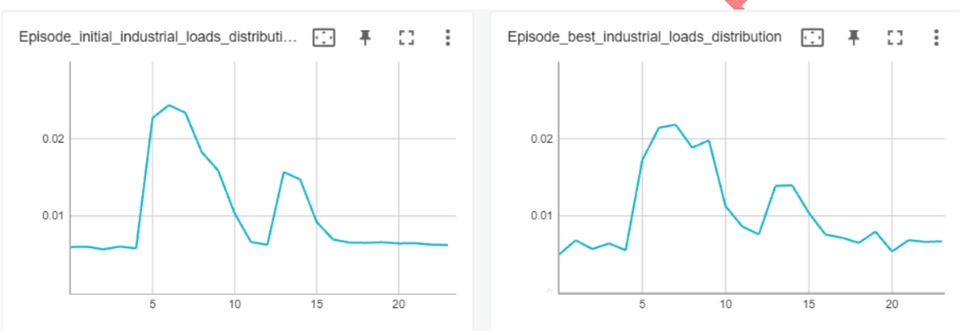


Figure 83: Industrial demand profile before and after the training process.

The optimization model requires further experimentation and performance tuning to gain higher EG rewards. This will be done in the next research iteration and reported in D7.4.

3.4.1.8.4 Demand Response integration

In this scenario, P2P energy trading between DSO and energy users in an automated, decentralised and flexible way. For each realization the following functionalities will be provided and validated through IoT-NGIN:

- Digital Twin services, providing context information about the status of the Grid and the IT network. It will thus provide to the DSO information about:
    - the generation and consumption levels which will be useful to deciding on initiating a DR auction or not.
    - potential vulnerabilities in the smart grid monitoring devices connected to the network, as well potential network attacks, which could compromise the operation of the DSO IT and OT processes.

  These are provided through the *IoT Device Indexing (IDI)* and *IoT Device Access Control (IDAC)* components, as well as the *IoT Vulnerability Crawler (IVC)* and the *Malicious Attack Detector (MAD)* cybersecurity tools.

- Demand Response (DR) marketplace API, which enables the DSO to create a DR auction-based request.

For the realization of this scenario, power grid status is monitored through a set of 43 smart meters, which provide power and voltage measurements. The data are injected to IoT-NGIN through the architecture of Figure 77 and Figure 78.

A sample of the data generated by the smart meters (sensors), as visualized in ASM's PROMOTIC SCADA system[8] is depicted in Figure 84. The properties' IDs are constructed following the OBIS code, which identifies the corresponding device value, according to the OBIS standard (IEC 62056-61[9])

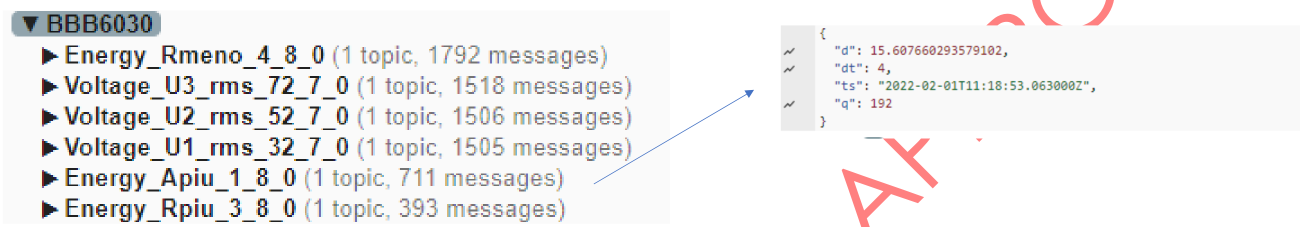


Figure 84: A sample of smart meter data.

Specifically, the monitored properties include:

- Energy_Rmeno_4_8_0= Negative reactive power (Q-) [kvar]
- Voltage_U3_rms_72_7_0= Instantaneous voltage (U) in phase L3 [V]
- Voltage_U2_rms_52_7_0= Instantaneous voltage (U) in phase L2 [V]
- Voltage_U1_rms_32_7_0= Instantaneous voltage (U) in phase L1 [V]
- Energy_Apiu_1_8_0= Positive active power (A+) [kW]
- Energy_Ameno_1_8_0= Negative active power (A-) [kW]
- Energy_Rpiu_3_8_0= Positive reactive power (Q+) [kvar]

The IDI component can be easily set up and integrated for any type of device or scenario. In this subsection, the instantiation for a use-case scenario under the scope of the Smart Agriculture Living Lab UC4 is presented in detail. However, the use of the component in other use cases is similar, so further analysis is not included for other use cases.

For the injection of smart meter data to the IDI component, a *Service* in IDI's IoT Agent (as explained in D4.3 [22]) must be created, in order to have the devices authenticate themselves when sending measurements. Next, the IoT Agent needs to register the devices

[8] MQTT Explorer, available online at: http://mqtt-explorer.com

[9] IEC 62056-61:2002 Object identification system (OBIS)

to the IDI's Context Broker. The smart meter is described by the following features, including the monitored properties of the smart meters.

- *smart_meter_id*, corresponding to the smart meter ID, as appearing in ASM's SCADA system.
- *type*, indicating the type of the specific device, which is *smart_meter* in this case.
- *ip_address*, which provides the IP address of the smart meter.
- *measurements*, corresponding to measurements of the monitored properties of the smart meters.

The smart meters' context data can be easily visualized by querying the API of IDI's Context Broker or IDI's Data Registry. Within UC9, the second option has been adopted for visualization.

In order to retrieve the smart meter data information, the multi-tenancy headers of IDI have to be set, as shown in Table 47.

Table 47: Device indexing multitenancy headers for use case #9.

	Fiware-Service	Fiware-ServicePath
Smart meter	meter	/

Regarding the validation of the cybersecurity services, this refers to the communication network.

Here, the IoT Vulnerability Crawler (IVC) will perform vulnerability scanning for the smart meters, while the Malicious Attack Detector (MAD) analyses network traffic and identifies anomalies, which can be considered as potential attacks.

In UC9 the network traffic data are collected by the pfSense firewall [24]. Specifically, the IT infrastructure of ASM has pfSense installed to manage the local network of 50 near real-time IoT Smart Meters. The related Attack Detector is SURICATA [25]. It has been decided to install pfSense in a server deployed to control and manage a bunch of Smart Meters installed to monitor 50 domestic customers in an apartment building. The firewall has been installed in a server that acts as a concentrator; the server has the following characteristics:

- Operating System: Windows 7 Professional
- RAM: 4 GB
- CPU Intel Core i3-4150 @ 3.5 GHz
- Hard disk: 500 GB

Deployment details are available in Figure 85 and Figure 86.

Figure 85: Smart meters for residential end users and the computer managing the connections.

Figure 86: The physical connection detail of the smart meters to the computer.

The firewall instance has been configured in order to gather data in a syslog server. The main acquired logs from the firewall are as follows:

- System logs: These logs provide general system-level information, including system messages, boot messages, and DHCP lease information.
- Firewall logs: These logs show firewall activity, including blocked and allowed traffic, and also include information about NAT and port forwarding.
- VPN logs: These logs provide information about VPN connections, including connection attempts and errors.
- DHCP logs: These logs provide information about DHCP leases and renewals.
- Captive portal logs: These logs show information about users who have accessed the captive portal.
- DNS logs: These logs provide information about DNS queries and responses.
- Proxy logs: These logs provide information about web proxy activity.
- Traffic shaper logs: These logs show information about traffic shaping rules and activity.

The main acquired logs from the attack detector are as follows:

- EVE JSON log: This is the primary log file generated by SURICATA. It contains detailed information about each network event that SURICATA detects, including metadata, packet information, and alerts.
- Fast log: This log provides a more condensed view of network traffic, suitable for high-speed traffic analysis.
- HTTP log: This log records HTTP transactions and can be used to identify suspicious web traffic.
- TLS log: This log records TLS sessions and can be used to identify suspicious SSL/TLS traffic.
- DNS log: This log records DNS queries and responses and can be used to identify suspicious domain names or DNS activity.
- File log: This log records information about files that are transferred over the network, including file name, size, and hash values.
- Stats log: This log records statistics about SURICATA's performance and can be used for monitoring and troubleshooting purposes.

Some example logs are in Figure 87.

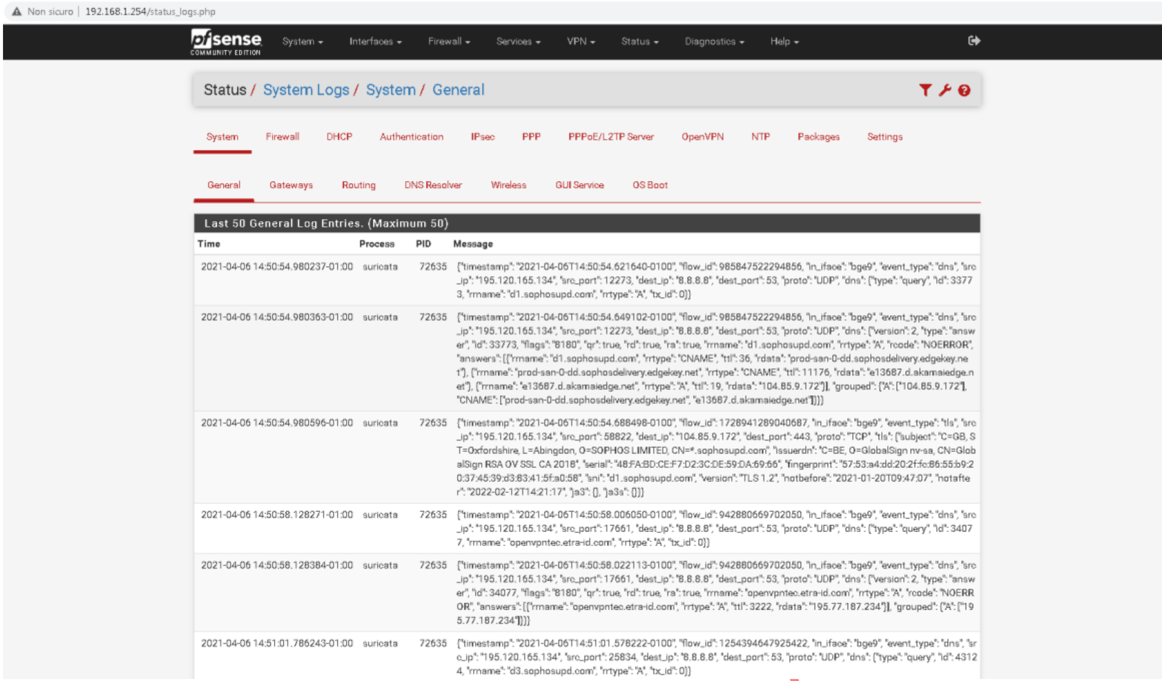


Figure 87: Log files produced by pfSense.

For the purposes of UC9, the syslog data file (in particular the EVE JSON log) will be dynamically read by MAD, in order to analyze the network data in almost real time.

3.4.1.8.5 Visualization dashboard

In this section we report some screen shots to show part of the results tested as described above in the paragraph "3.4.1.2. Testing Scenarios".

3.4.1.8.5.1 Dashboard: Home web page

(At the moment some data are not yet available)

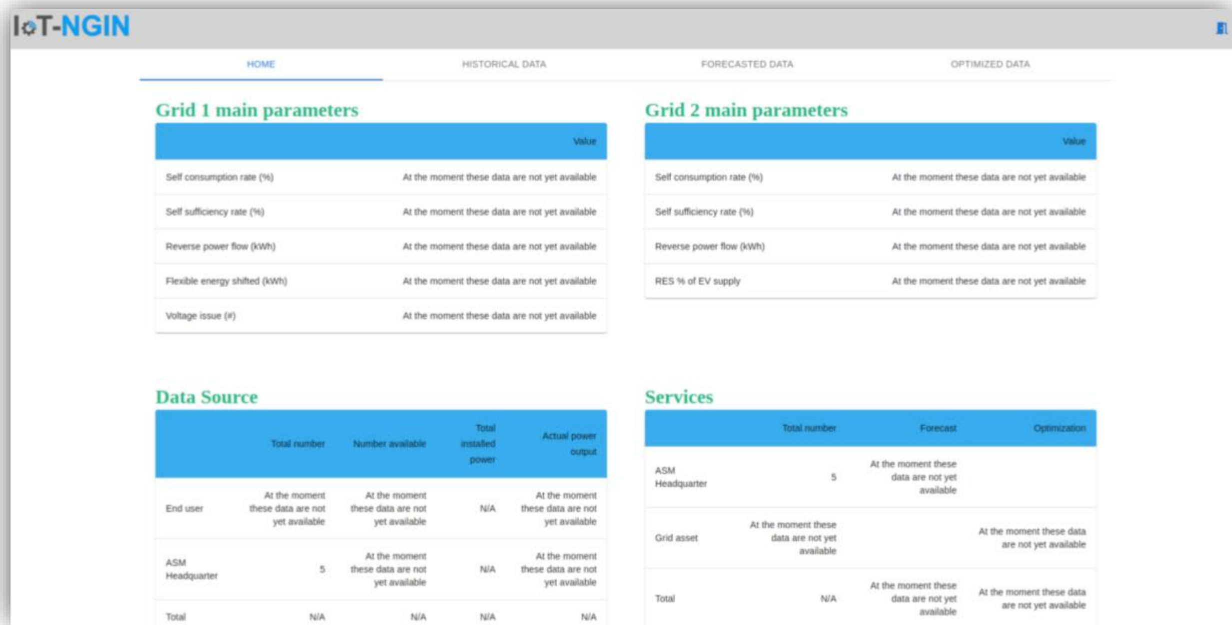


Figure 88: Dashboard - Home web page.

3.4.1.8.5.2 Dashboard: Historical data (ASM HQ)

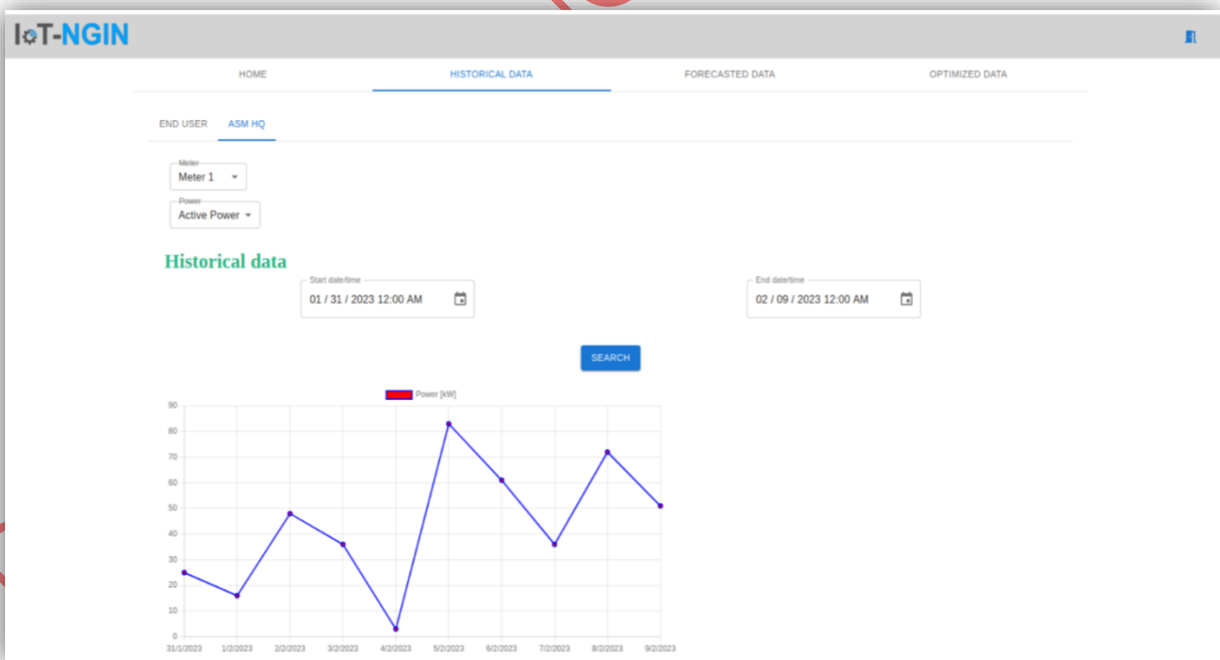


Figure 89: Dashboard - Historical data.

3.4.1.8.5.3 Dashboard: Historical data (End User)

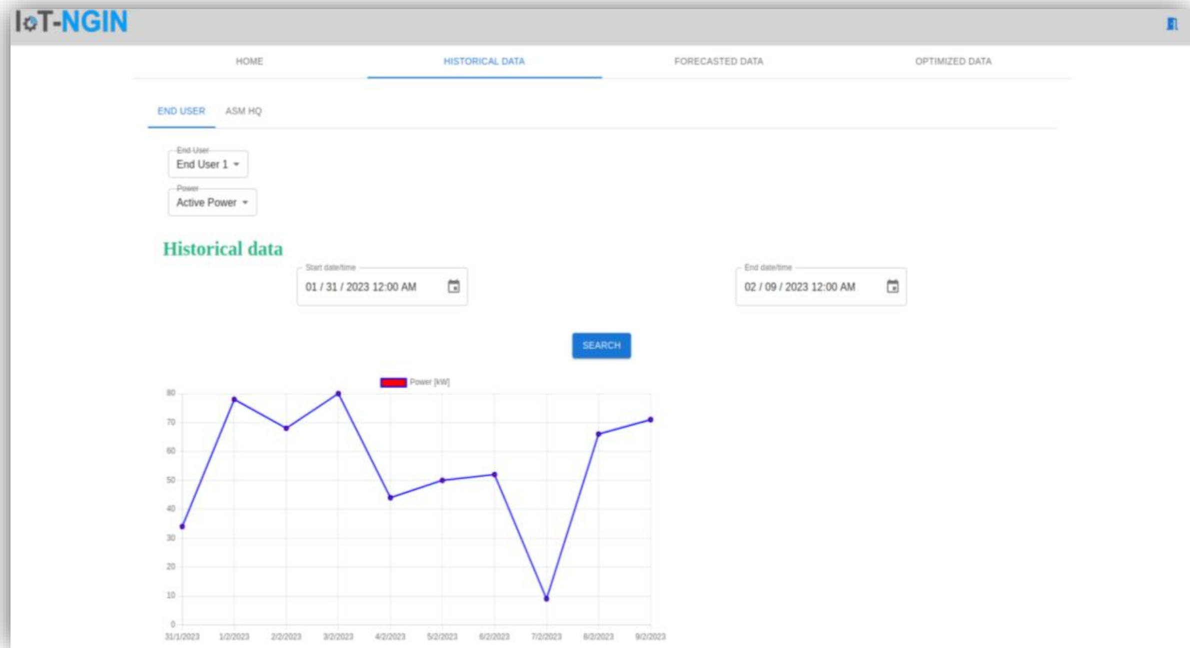


Figure 90: Dashboard - Historical data (end user).

3.4.1.8.5.4 Dashboard: Forecasted data

(At the moment the data are not yet available)

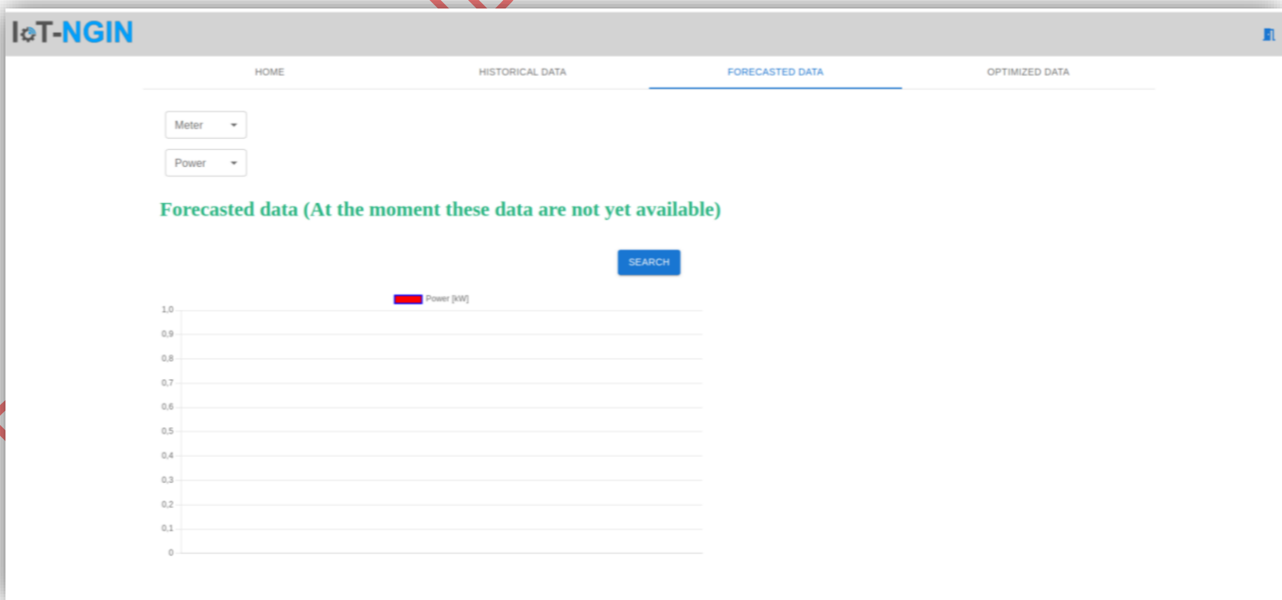


Figure 91: Dashboard - Forecasted data.

3.4.1.8.5.5 Dashboard: Optimized trend

(At the moment the data are not yet available)

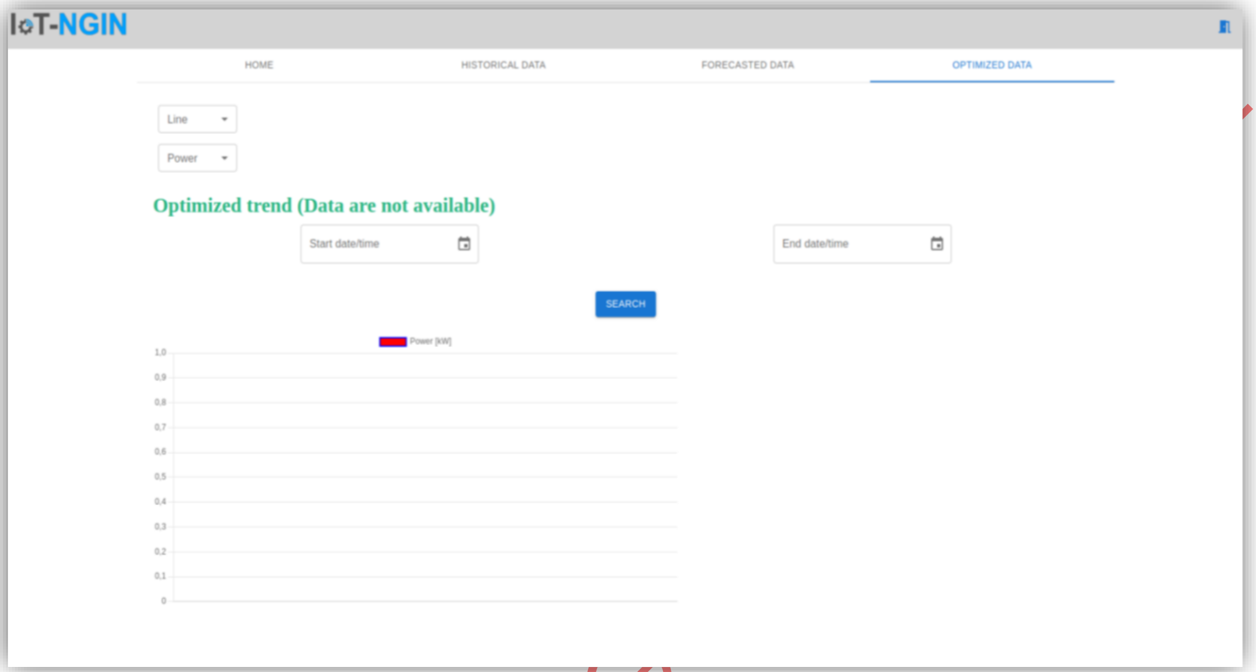


Figure 92: Dashboard - Optimized trend.

The status of the dashboard and future activities to be performed are collected in the following table:

Table 48: The status of the dashboard.

Activity	Description	Equipment	Status
Sharing of real -time data on dedicated dashboard interface	Real -time data can be obtained for a precise list of sensors provided to be shown on the dashboard, subscribing to the Mqtt Broker of ASM (after authentication and access via VPN)	Server ASM (and Sensor) MQTT Broker of ASM	Available
Sharing historical data on dedicated dashboard interface	Historical data can be obtained for a precise list of sensors provided to be shown on the	Server ASM (and Sensor) database SQL	Available

	dashboard, accessing the SQL database (after authentication and access via VPN)		
Sharing data forecast generation and consumption on dedicated dashboard interface	The forecast data can be obtained for a precise list of sensors provided to be shown on the dashboard, accessing the ASM server via Exposed Prediction Service (after authentication and access via VPN)	Server ASM (and Sensor)	Not available yet
Optimized data sharing data On a dedicated dashboard interface Data relating to the optimization of the operations and simulation of the benefits deriving from the adoption of service restoration strategies.	The forecast data can be obtained for a precise list of sensors provided to be shown on the dashboard, accessing the ASM server (after authentication and access via VPN)	Server ASM (and Sensor)	Not available yet

3.4.2 UC10 – Driver-friendly dispatchable EV charging

IoT-NGIN Use Case #10 "Driver-friendly dispatchable EV charging" aims to address economic and environmental challenges, which are strengthened by the simultaneous transitioning to renewable energy and electric mobility.

In particular, the increase of electric vehicles causes an increment of electricity that must be supplied and, therefore, a necessary strengthening of power lines follows; moreover, based on European Commission Directive 2018/2001/EU [26], this electricity will progressively come from intermittent and non-programmable renewable energy plants, resulting in an energy balancing challenge. In this context, a cooperation mechanism between DSO (Distribution System Operator), CPO (Charging Point Operator) and EV users allows a power lines improvement limitation providing grid balancing service by coordinating EV charging. DSO monitors the electricity grid and, thanks to accurate forecasting systems, based on ML models, will be able to identify how, when and where to charge electric vehicles for grid balancing service, managed by blockchain-based marketplace that enables P2P energy trading between DSO and CPO via smart contracts and micro payments, using distributed IoT devices for real-time verification of energy flexibility service provision.

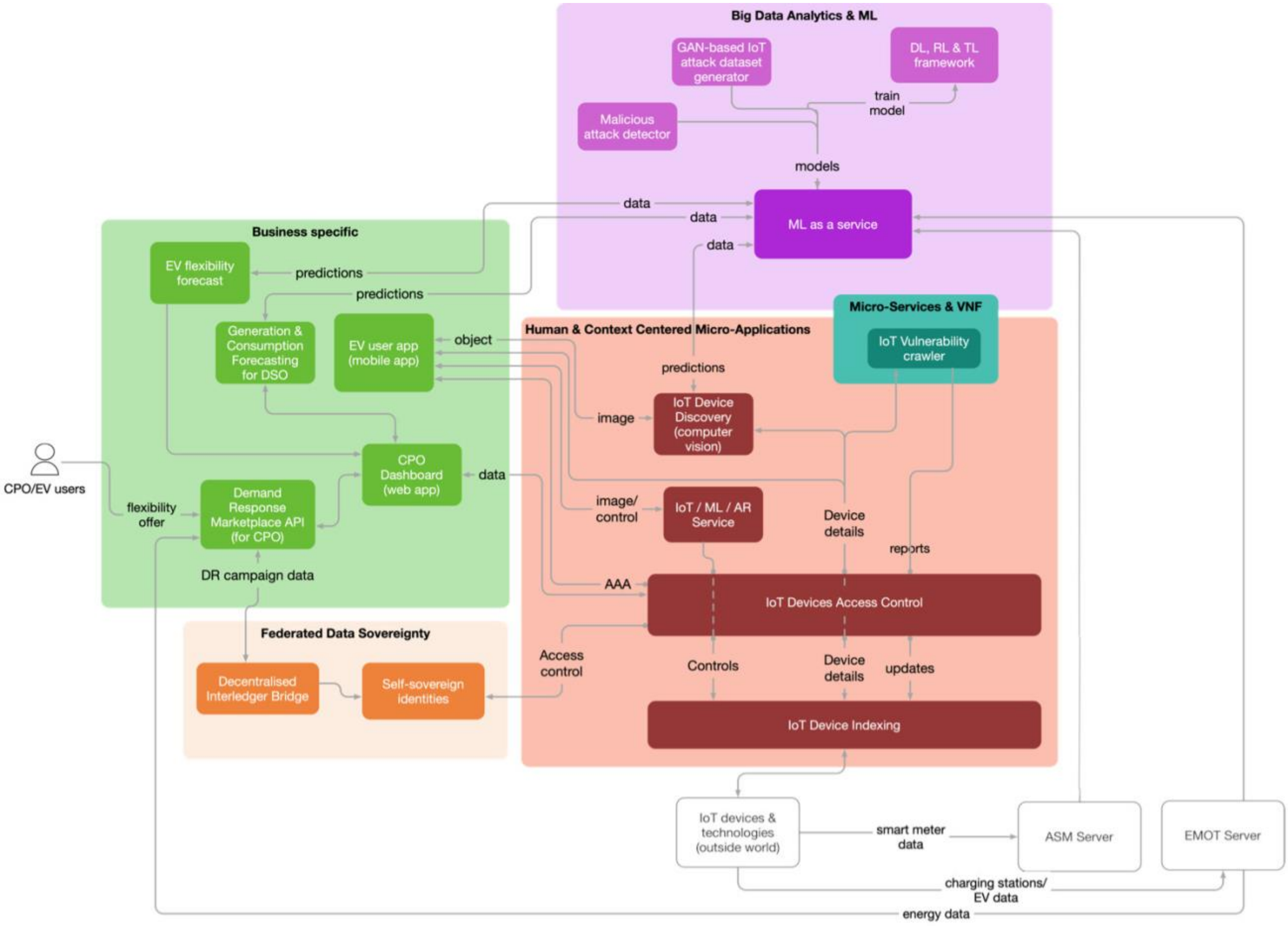


Figure 93: UC10 architecture instantiation.

3.4.2.1 Trial site description

The trial is conducted in ASM Living Lab at Terni, Italy, in which four substations, a 240 kWp PV plant, six electric vehicles and three charging stations has been involved. The pilot infrastructure is shown below in Figure 94.

Figure 94: ASM Terni Living Lab.

3.4.2.2 Required equipment

Table 49 below provides the detailed description of the required equipment for UC10, as already reported in D7.2, along with its type and status.

Table 49: Detailed description of the required equipment for UC10.

Equipment	Description / Specifications	Type	Status
ASM server	It's a SCADA system made by Wonderware, able to monitor in real time hundreds of sensors and to store historical data.	Server	Available
EMOT server	Server with 4 core CPU + 8 GB RAM	Server	Available
Smart Meter	About 100 electrical smart meters, able to monitor active and reactive power and voltage in several buses of the grid	Sensor	Available
PMU	2 phasor measurement units, able to monitor active and reactive power, voltage, phase angle, harmonic content, etc.	Sensor	Available

PQA	6 power quality analysers, able to monitor active and reactive power, voltage, harmonic content, etc.	Sensors	Available
Electric Vehicle	4 x Renault ZOE, 2 x Nissan LEAF	Vehicle	Available
OBD	OBD is a IoT component; OBD connects to the diagnostic interface from which it can extract the information from the electric vehicle control unit	Sensor	Available
Charging Station	3 x Emotion EVO (22 kW), 1 x Emotion FAST (50 kw)	Hardware	Available
Raspberry Pi 3	Single-board computer installed inside of the charging stations to enable real-time monitoring and remote management	Hardware	Available
CPO Platform	Electric Vehicle and Charging station Monitoring and Management Platform	Software	Available
Camera	Camera to perform object detection	Sensor	Available
Smartphone	Smartphone to perform augmented reality interaction with charging station	Hardware	Available
DSO Platform	Living Lab Grid Monitoring and Management Platform	Software	Available

3.4.2.3 Data collection

The following dataset(s) have been identified as part of the use case:

- **Power Quality Analysers (PQA) dataset:** Measurements of voltage, currents, and power derived by ASM energy units, collected through HTTP protocol only via LAN/VPN connections, which provide built-in security access features.
- **Charging station data:** Real-time and historical data collected from the charging stations deployed in the trial site (Terni, Italy). The data will include information about the charging station and each charging session, without any identifying information about the use of the station.
- **Electric vehicle data:** Real-Time and historical data collected from electric vehicles deployed in the trial site (Terni, Italy). The data will include battery capacity, power and life, and data about the vehicles' movement and trajectories, such as latitude, longitude, and speed.

3.4.2.4 Alignment with IoT-NGIN technologies

Table 50: Alignment of UC10 with the relevant IoT-NGIN technologies.

Task 3.1 – MLaaS: Grid Forecasting	
Description	Data collected from smart meters will be used to train machine learning, obtaining a grid status forecasting system able to provide day-ahead prediction about DSO energy flexibility needs.
Adaptation and fine-tuning	The technology will be developed using specific machine learning tools to obtain an action strategy for grid balancing, i.e., an indication of how much energy flexibility is needed (kWh) and at what time of day.
Deployment	The service will be carried out on the cloud and will be based on the data that is communicated by ASM.
Related requirements	**REQ_SE2_F10** – DSO shall be able to forecast electricity production / consumption and estimate flexibility need
Related KPIs	**KPI_UC10_3** – Reduction of reverse power flows ~ 10 kWh/day **KPI_UC10_4** – Increase EV charging efficiency, money saved: 0.05 €/kWh **KPI_UC10_5** – Increase EV charging efficiency, renewable energy > 50%
Task 3.2 – MLaaS: EV Flexibility Provision Forecasting	
Description	Data collected from electric vehicle OBDs will be used to train machine learning, obtaining a forecasting system able to provide day-ahead prediction about EV potential energy flexibility provision.
Adaptation and fine-tuning	The technology will be developed using specific machine learning tools to obtain an action strategy for flexibility provision, i.e. an indication of how much energy flexibility will be available (kWh) at DSO needed time.
Deployment	The service will be carried out on the cloud and will be based on the data that is communicated by EMOT.
Related requirements	**REQ_SE2_F07** – Charging station must provide energy data to be involved in DR campaigns; data shall be stored for result evaluation
Related KPIs	**KPI_UC10_3** – Reduction of reverse power flows ~ 10 kWh/day **KPI_UC10_4** – Increase EV charging efficiency, money saved: 0.05 €/kWh **KPI_UC10_5** – Increase EV charging efficiency, renewable energy > 50%
Task 4.2 – Device Discovery & Indexing Module	

Description	Device discovery and indexing module will be used to allow charging station detection via a camera installed on the electric vehicles.
Adaptation and fine-tuning	Computer vision models will be trained over charging station videos provided by EMOT.
Deployment	Cloud
Related requirements	**REQ_SE2_F02** – e-Mobility platform shall be enabled for user registrations **REQ_SE2_F05** – Charging station must be connected to the internet to be integrated into the platform **REQ_SE2_F06** – Electric vehicle must be connected to the internet to be integrated into the platform
Related KPIs	**KPI_UC10_4** – Increase EV charging efficiency, money saved: 0.05 €/kWh **KPI_UC10_5** – Increase EV charging efficiency, renewable energy > 50%
T4.3 – Device Access Control Module	
Description	Device Access Control Module will enable augmented reality interaction between charging station and electric vehicle, after the charging station detection is done.
Adaptation and fine-tuning	Device Access Control Module provides access management to EMOT e-Mobility platform
Deployment	Cloud
Related requirements	**REQ_SE2_F02** – e-Mobility platform shall be enabled for user registrations **REQ_SE2_F05** – Charging station must be connected to the internet to be integrated into the platform **REQ_SE2_F06** – Electric vehicle must be connected to the internet to be integrated into the platform
Related KPIs	**KPI_UC10_4** – Increase EV charging efficiency, money saved: 0.05 €/kWh **KPI_UC10_5** – Increase EV charging efficiency, renewable energy > 50%
Task 4.4 – AR Tool	
Description	AR tool will enable novel interaction between electric vehicle users and charging point operator, allowing charging session management in AR mode.
Adaptation and fine-tuning	AR tool will be adapted for smartphones application

Deployment	Cloud
Related requirements	**REQ_SE2_F02** – e-Mobility platform shall be enabled for user registrations **REQ_SE2_F05** – Charging station must be connected to the internet to be integrated into the platform **REQ_SE2_F06** – Electric vehicle must be connected to the internet to be integrated into the platform
Related KPIs	**KPI_UC10_4** – Increase EV charging efficiency, money saved: 0.05 €/kWh **KPI_UC10_5** – Increase EV charging efficiency, renewable energy > 50%
Task 5.1 – IoT vulnerability crawler	
Description	This component will identify potential vulnerabilities in EV charging stations
Adaptation and fine-tuning	No adaptation need identified at this point.
Deployment	Edge/Cloud
Related requirements	**REQ_SE2_NF06** – Data shall be consistent, reliable, transparent and accessible only to authorized users
Related KPIs	**KPI_UC10_3** – Reduction of reverse power flows ~ 10 kWh/day **KPI_UC10_4** – Increase EV charging efficiency, money saved: 0.05 €/kWh **KPI_UC10_5** – Increase EV charging efficiency, renewable energy > 50%

Task 5.4 –Self-Sovereign Identities and Interledger applications	
Description	SSIs will enable pervasive security to the EV charging stations control access while decentralized interledaer bridge will enable information exchanging among different DLTs to storage key information related to DR campaign from a light DLT into a more secure DLT.
Related requirements	**REQ_SE2_NF06** – Data shall be consistent, reliable, transparent and accessible only to authorized users
Related KPIs	**KPI_UC10_3** – Reduction of reverse power flows ~ 10 kWh/day **KPI_UC10_4** – Increase EV charging efficiency, money saved: 0.05 €/kWh **KPI_UC10_5** – Increase EV charging efficiency, renewable energy > 50%

Task 5.1 – Attack detection (Malicious Attack Detection and GAN based attack)

Description	Malicious attack detection at the network level, applied to the data management platform of IoT-NGIN
Adaptation and fine-tuning	Systems will be used to detect attacks and protect the platform from a cybersecurity perspective.
Deployment	Cybersecurity services produced within the data management platform will be tested
Related requirements	**REQ_SE2_NF06** – Data shall be consistent, reliable, transparent and accessible only to authorized users
Related KPIs	**KPI_UC10_3** – Reduction of reverse power flows ~ 10 kWh/day **KPI_UC10_4** – Increase EV charging efficiency, money saved: 0.05 €/kWh **KPI_UC10_5** – Increase EV charging efficiency, renewable energy > 50%

3.4.2.5 Testing Scenarios

For validating the IoT-NGIN efficiency towards providing added value in Smart Energy in relation to AR-based charging management, energy flexibility prediction and provision, three test scenarios have been defined and are being implemented in the LL with the use of the IoT-NGIN tools. The test scenarios are described in the following subsections.

3.4.2.5.1 ML-based energy flexibility prediction and P2P energy trading via blockchain marketplace

Table 51: UC10 Test 1 "ML-based flexibility prediction and P2P energy trading via blockchain marketplace".

Test 1: ML-based flexibility prediction and P2P energy trading via blockchain marketplace	
Objective	The objective of this test is to predict energy flexibility needs to maintain grid stability in a condition of distributed renewable energy plants high penetration, obtaining a reduction in energy costs and CO_2 emissions. Forecasted energy flexibility needs will trigger a DR campaign using EVs as energy flexibility provider; blockchain-based marketplace enables P2P energy trading between DSO and CPO, acting also as EV aggregator.
Components	• IoT Device Indexing • IoT Device Access Control • MLaaS • Malicious Attack Detector (MAD) • GAN-based dataset generation • Privacy-preserving Self-Sovereign Identities (SSIs) • IoT vulnerability crawler

Features to be tested	• ML-based forecasting service for energy consumption at grid level development
	    o Training
	    o Deployment
	    o Inference
	• ML-based forecasting service for energy production at grid level development
	    o Training
	    o Deployment
	    o Inference
	• ML-based forecasting service for EV flexibility provision development
	    o Training
	    o Deployment
	    o Inference
	• Smart Energy Digital Twin
	• Blockchain-based Marketplace
	• Access control
	• Grid platform
	• Electric mobility platform

Requirements addressed	REQ_SE2_F01, REQ_SE2_F02, REQ_SE2_F03, REQ_SE2_F04, REQ_SE2_F05, REQ_SE2_F06, REQ_SE2_F07, REQ_SE2_F09, REQ_SE2_F10, REQ_SE2_F11, REQ_SE2_F12, REQ_SE2_F13, REQ_SE2_NF01, REQ_SE2_NF02, REQ_SE2_NF03, REQ_SE2_NF04, REQ_SE2_NF05, REQ_SE2_NF06, REQ_SE2_NF07
Test setup	The IoT-NGIN framework should be installed and functional. The IoT-NGIN-enabled smart meters, charging stations and electric vehicles should be prepared. DSO user should be provided with permissions to access smart meters and grid platform. CPO user should be provided with permissions to access charging stations, electric vehicles and electric mobility platform. The smart meters, charging stations and electric vehicles are integrated with their Digital Twin at the edge device, realized via IDI and IDAC and keeping its data acquired on a regular basis, but it can also receive control commands through it.
Steps	1. DSO user triggers the training of a new ML model for energy consumption/production at grid level prediction 2. CPO user triggers the training of a new ML model for EV flexibility provision prediction 3. DSO user triggers the deployment of the new model into the grid platform 4. CPO user triggers the deployment of the new model into the electric mobility platform 5. DSO user triggers the deployment of the blockchain-based marketplace into the grid platform 6. CPO user triggers the deployment of the blockchain-based marketplace into the grid platform 7. DSO creates DR campaign based on grid forecasted data 8. CPO provides offer for DR campaign based on EV forecasted data 9. Smart contract is signed 10. EV users provide energy flexibility according to DR campaign rule 11. Micro-payment in token is provided by DSO to CPO for flexibility provision and smart contract is closed

KPIs	KPI_UC10_1, KPI_UC10_2, KPI_UC10_3, KPI_UC10_4, KPI_UC10_5
IoT-NGIN innovations	• Smart meters, charging stations and electric vehicles data and predictions are available on IoT-NGIN's DT, protected by Access Control • Efficient use of resources and minimization of communication costs, as ML inference takes place at the far edge node, while model training happens at the edge – no need to send data to the cloud – edge analytics • Continuous enhancements to the prediction efficiency are possible, through a simple and automated process for ML model training

3.4.2.5.2 AR-based charging management

Table 52: UC10 Test 2 "AR-based charging management".

Test 2: AR-based charging management	
Objective	The objective is to allow AR-based charging station management for future application with eye smart glasses and car smart glasses.
Components	• IoT Device Discovery (IDD) • IoT Device Indexing (IDI) • IoT Device Access Control • AR module • MLaaS • AR Mobile app
Features to be tested	• Development of Charging Station's Digital Twin, realized through the IDI component • Data acquisition from charging station Digital Twin • ML-based Image recognition service • Pervasive security in the context of ambient intelligence, granting device access based on user credentials, location information and device recognition • AR visualization of charging station info on user's mobile device • AR-based actuation: charging session start/stop and charging station power output modulation
Requirements addressed	REQ_SE2_F01, REQ_SE2_F02, REQ_SE2_F03, REQ_SE2_F05, REQ_SE2_NF01, REQ_SE2_NF02, REQ_SE2_NF03, REQ_SE2_NF04
Test setup	The IoT-NGIN framework should be deployed and functional. A set of charging station is installed and configured to communicate with the edge node and the electric mobility platform. The electric mobility platform must be integrated with IDI, in order to inject energy real-time measurements, but also receive actuation commands. AR Mobile App is installed on the user's device and the user is eligible to access data of at least one of the installed charging stations, protected by the IDAC module. The Computer Vision (CV)-based image recognition module of the IDD component should be deployed and accessible via the MLaaS platform, based on already trained ML model, able to recognize

	charging stations. Each charging stations carries a QR code, which provides the charging station's id information.
Steps	1. EV user takes a video of charging station using smartphone camera 2. Video is sent to MLaaS for charging station detection (computer vision) 3. After charging station has been discovered, charging station is accessed via access control module and charging station info are provided via device indexing module 4. EV user configures charging session using AR and AR module sends configuration to electric mobility platform
KPIs	KPI_UC10_1, KPI_UC10_2
Innovations brought by IoT-NGIN	• Monitoring data are available on IoT-NGIN's DT, protected by Access Control • Any calculation takes place at the edge – no need to send data to the cloud – edge analytics • Access to charging stations is protected and controlled through multiple criteria, selected for the use case • Enhanced interaction with charging stations

3.4.2.6 Use case sequence diagrams

The updated use case diagrams for UC10 are presented in the following. Figure 95 presents the sequence diagram for ML-based energy flexibility prediction and P2P energy trading via blockchain marketplace.

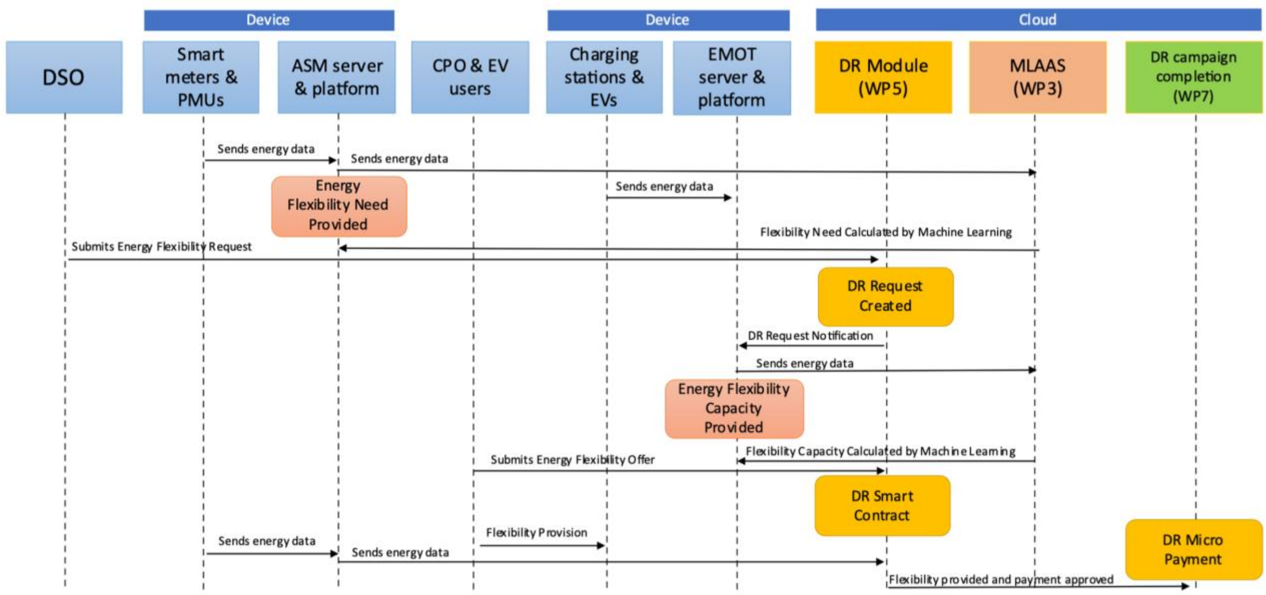


Figure 95: Sequence diagram for ML-based energy flexibility prediction and P2P energy trading via blockchain marketplace.

Figure 96 presents the sequence diagram for AR-based charging management.

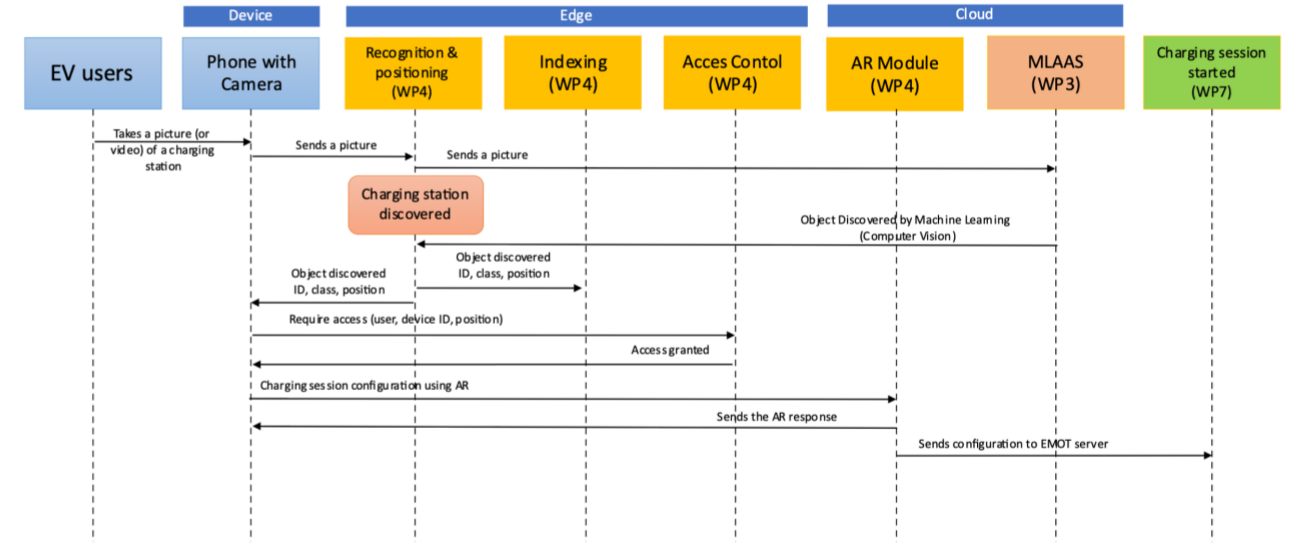


Figure 96: Sequence diagram for AR-based charging management.

3.4.2.7 Execution timeline

The pilot activities are conducted, according to the plan presented in Table 53, which illustrates that the third wave of the LL activities, concerning the intermediate implementation and validation of the IoT-NGIN tools, has been completed. The LL validation activities, against the three defined scenarios, will be running until the end of the project.

Table 53: Execution timeline of UC10.

Phase	Estimated start date	Estimated end date	Notes
Trial set-up and equipment procurement	M8	M15	*Completed*
Initial implementation and validation	M16	M20	*Completed*
Intermediate implementation and validation	M21	M26	*Completed*
Final implementation and validation	M27	M36	Running

3.4.2.8 Intermediate results

The pilot activities for UC10 so far have been focused on:

- Data collection through integration of IoT-NGIN's digital twinning functionality with the smart meters, charging stations and electric vehicles
- IoT-NGIN framework deployment and validation on the Smart Energy LL

For the needs of the IoT-NGIN validation, IoT-NGIN tools need to be installed in the Smart Energy LL. Following the integration & deployment approach of the project, a K8S installation has been made available in the LL facilities and IoT-NGIN tools have been deployed. Specifically, the following components have been deployed on the Smart Energy LL:

- IoT Device Discovery
- IoT Device Indexing
- IoT Device Access Control
- MLaaS
- AR module
- Privacy-preserving Self-Sovereign Identities (SSIs)
- IoT Vulnerability Crawler
- Malicious Attack Detector.

In the following, the LL activities for each of the test scenarios under UC10 are briefly presented.

3.4.2.8.1 ML-based energy flexibility prediction and P2P energy trading via blockchain marketplace

For the execution of this scenario, one hundred smart meters, three charging stations and six electric vehicles has been integrated in IoT-NGIN pilots.

IoT devices data are used for consumption/production prediction, EV flexibility potential prediction and flexibility provision verification.

Consumption/production prediction and EV flexibility potential prediction services have been designed, implemented and tested in UC10 as a joint collaboration between WP6 T6.2 and WP3 T3.2, by leveraging the MLaaS Online Learning framework. They have been integrated as part of the WP3 MLaaS platform and its technical details preliminary reported in D3.3.

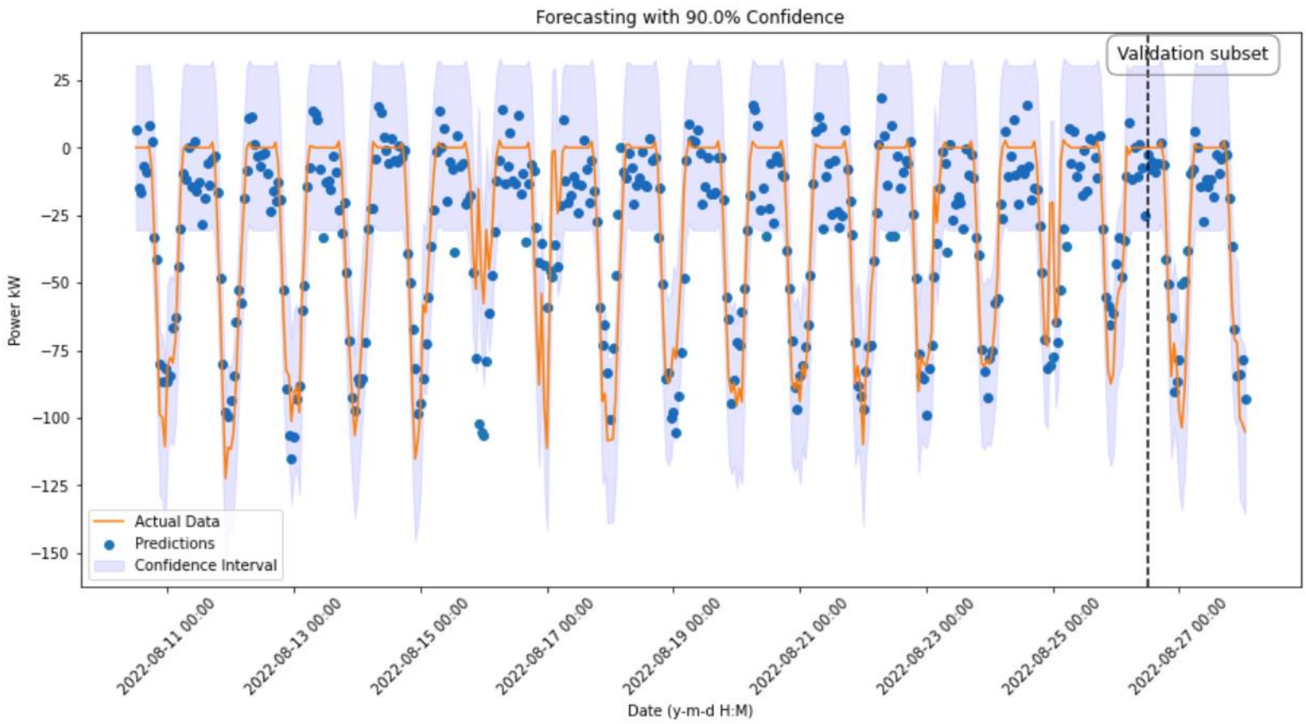


Figure 97: Predictions for power generation of UC10.

Smart meters data are available through the grid platform managed by DSO, as shown in UC9 section.

Charging stations and electric vehicles historical and real-time data has been shared with technical partners using MQTT and they are available through the electric mobility platform dashboard, as depicted in figures below.

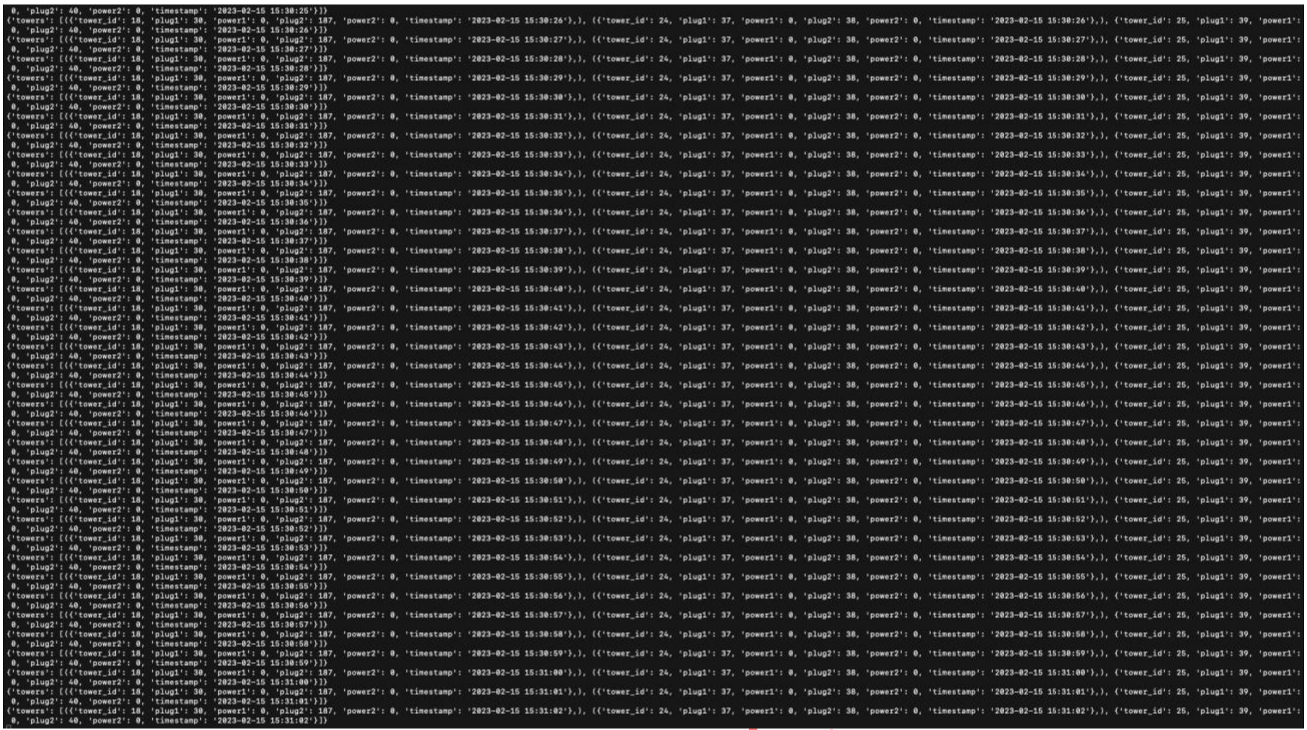


Figure 98: Smart Energy LL charging stations real-time data on EMOT MQTT broker.

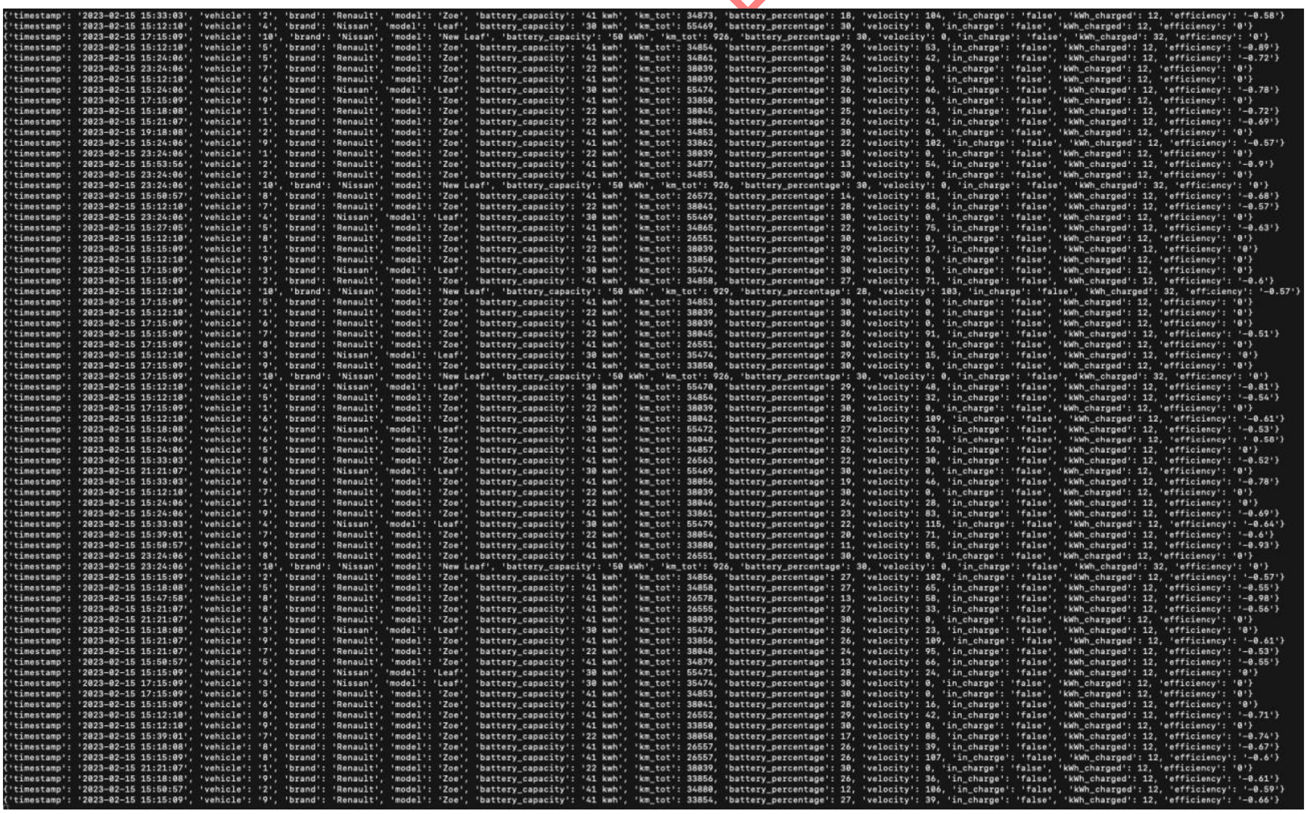


Figure 99: Smart Energy LL electric vehicles real-time data on EMOT MQTT broker.

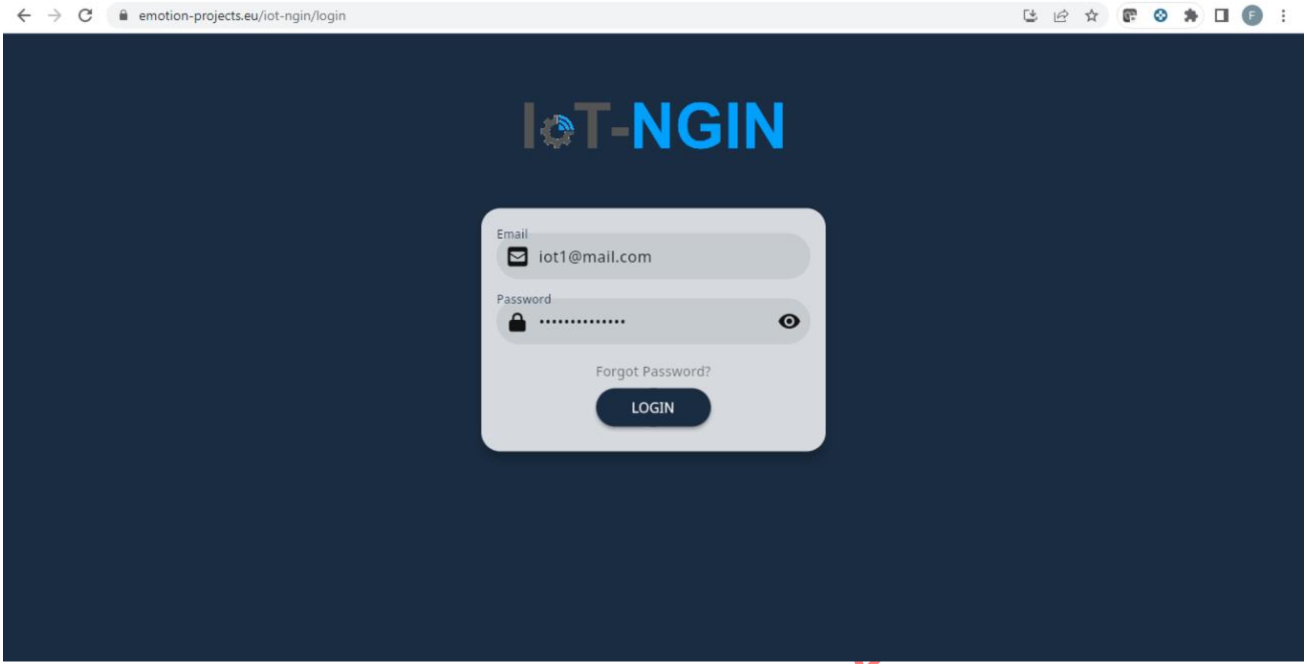


Figure 100: Electric mobility platform login.

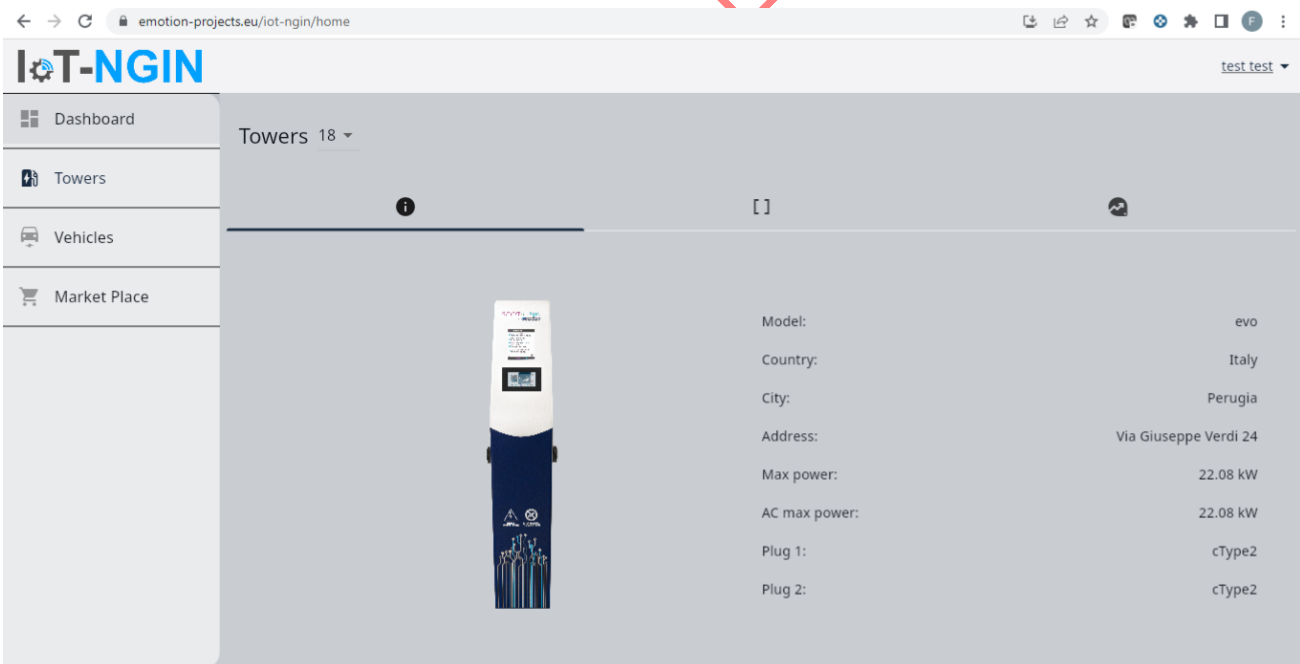


Figure 101: Smart Energy LL charging station panel in the electric mobility platform.

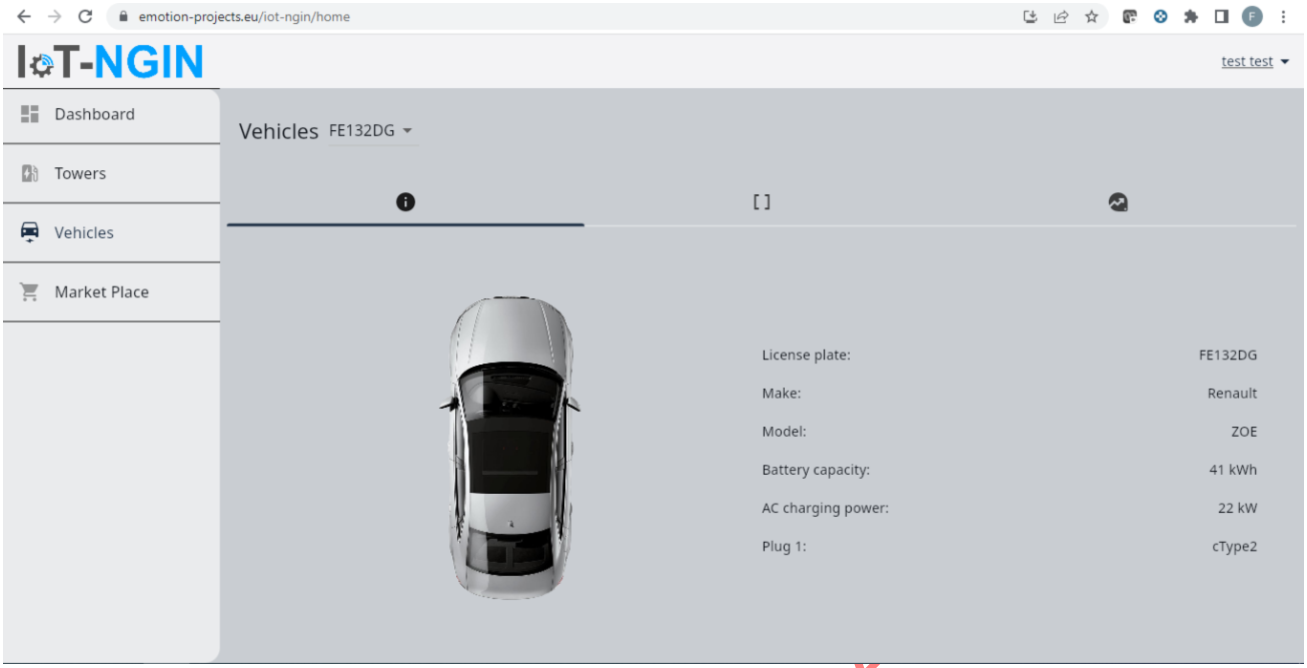


Figure 102: Smart Energy LL electric vehicle panel in the electric mobility platform.

Furthermore, blockchain-based marketplace developed in SOFIE project [27] has been integrated in electric mobility platform to enable P2P energy trading between DSO and CPO for Demand Response campaign. Through the marketplace, DSO can create an auction to obtain the energy flexibility service. CPOs involved in the marketplace can submit an offer in the marketplace and the most advantageous offer for the DSO is selected; at the same time a smart contract is created between the two actors which will be resolved when the service is provided, and a token transaction is released. The goal of the SOFIE Marketplace component is to enable the trade of different types of assets in an automated, decentralized and flexible way using Ethereum smart contracts. Figure 103 shows how an auction takes place using the Marketplace component [28].

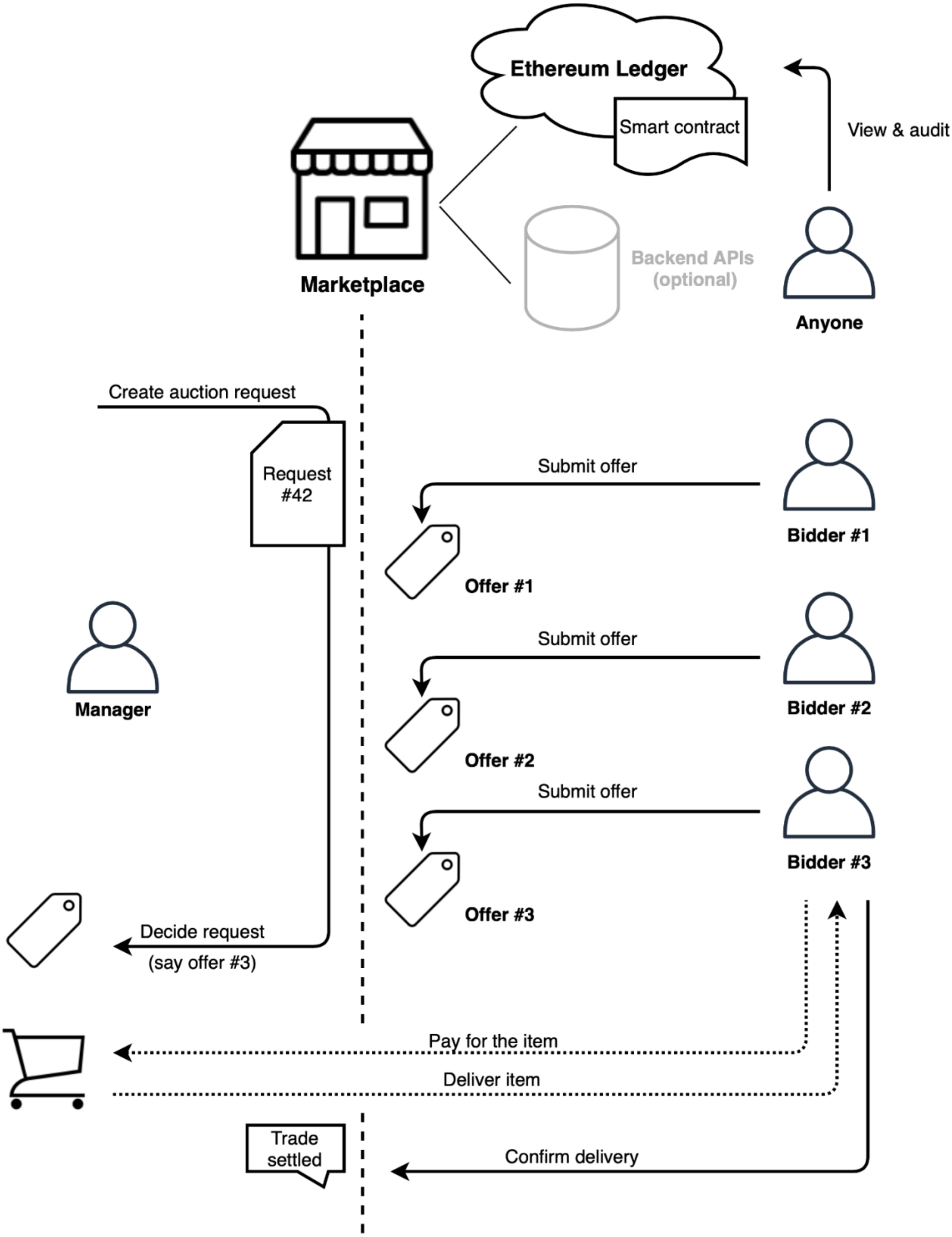


Figure 103: The flow of an auction using the Marketplace component.

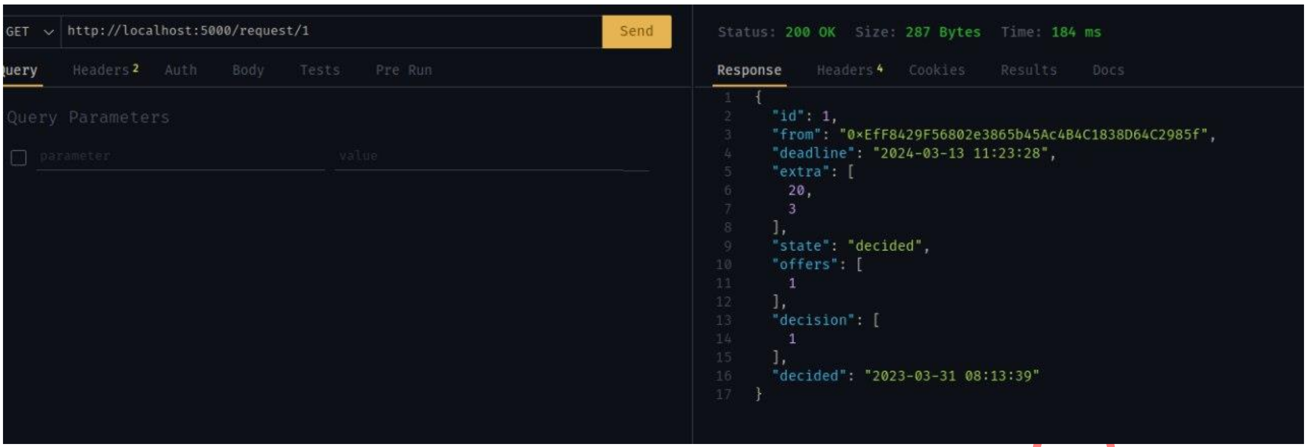


Figure 104: DSO flexibility request in blockchain-based marketplace.

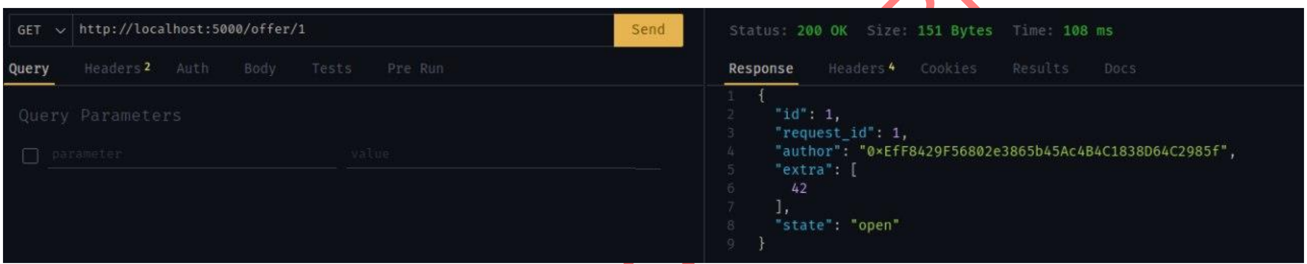


Figure 105: CPO flexibility offer in blockchain-based marketplace.

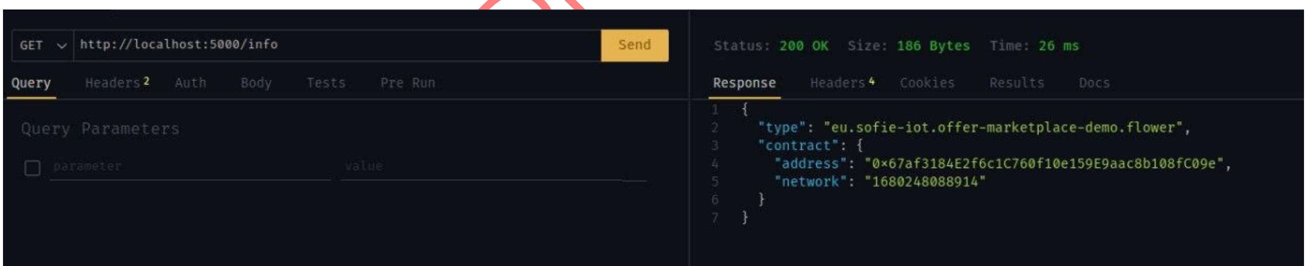


Figure 106: Demand Response smart contract in blockchain-based marketplace.

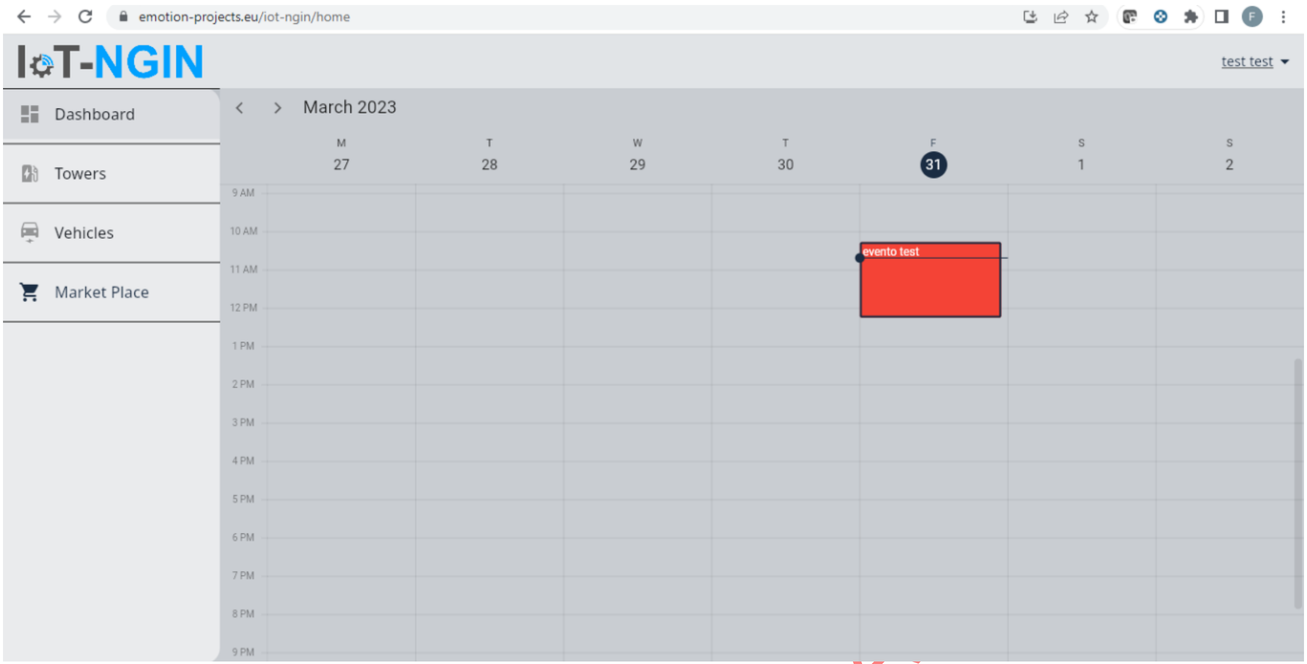


Figure 107: Marketplace calendar in electric mobility platform.

Since the cooperation mechanism between electricity grid operators and electric mobility operators is based on distributed IoT devices that interact with digital platforms to perform digital services, it is essential to guarantee cyber security to ensure data integrity and, therefore, the effectiveness of the energy flexibility service. For this reason, IoT Vulnerability Crawler, GAN based dataset generator and Malicious Attack Detection (MAD) components have been considered for ML-based energy flexibility prediction and P2P energy trading via blockchain marketplace scenario:

- IoT Vulnerability Crawler scans the UC10 network employed for IoT-NGIN demonstration activities.
- GAN based dataset generator produce datasets to support training of UC10 network anomaly detection algorithms for Malicious Attack Detection component.
- MAD detects network attacks and federated learning attacks for data poisoning.

Smart meters and their relative traffic network data have been used for the intermediate validation of IoT Vulnerability Crawler, GAN based dataset generator and Malicious Attack Detection (MAD) components as described in UC9 section, while charging stations and electric vehicles results will be reported in D7.4.

3.4.2.8.2 AR-based charging management

In Smart Energy Living Lab, UC10 is also considered to test IoT devices discovery, recognition and indexing and to specify a new way to control IoT devices, using augmented reality technology. In the UC10, electric mobility plays the role of a flexibility provider for the stabilization of the electricity grid heavily penetrated by distributed renewable energy plants. In this scenario, the charging station is the IoT device to be recognized and to be enabled for the interaction using AR technology; electric vehicles deployed in the Italian living lab, equipped with a camera, have been used to collect images of the charging stations at different times of the day and in different time conditions, so as to be able to guarantee learning by artificial intelligence system and obtain a reliable discovery and recognition mechanism. After learning process, it has been possible to start the implementation of AR interaction mode with the charging station, using the APIs of charging station monitoring and management platform developed by EMOT. In this way, EV user is able to view the details of the charging station, as well as access and control it in AR mode.

Table 54: AR-based charging management tests list.

N. TEST	TEST MACRO-SCENARIOS	Description	Equipment	Status
T01 (includes: T07+T06+T05+ T04+T03+T02)	Macro General Test including IoT Device Discovery & IoT Device Indexing + "Actuate Command"	This macro test illustrates the use of the IoT Device Discovery component which detects the charging station by training the ML tool (MLaaS Module). Then we will test that during the video shooting via an Android mobile phone. The recharging station is recognized and detected by showing on the display of the phone itself a percentage of recognition of the object detected during the video shooting and anchoring it on the real image, including also a colored bounding box (e.g. green) once recognized. Instead, if you move the mobile phone to another object other than the charging station (for example a tree, a car or other), this information described above (% object detected and anchor boxes) will not appear on the mobile phone display, testing in this way the recognition of the Device. The macro test continues by illustrating the use of IoT Device Indexing protected by IoT Device Access Control. IoT Device indexing services are accessed by making requests to IoT Device Access Control's Keycloak-protected endpoints. The test regarding the IoT Device Indexing module tests the following functionalities: - Retrieving a token via Keycloak	Server EMOT (MLaaS) + Server ASM (IoT Device Indexing) + Server ENG (AR Module)	Not available yet

		- Request for information relating to measurements for a specific device, identified via QR Code - Retrieve information about digital twins of devices. - AR mode activation by overlaying real image data on the mobile phone display, such as for example: Device ID, percentage Charge, etc. Then the discovery and indexing of the charging station will activate the control in AR mode. At the end the "Actuate command" is tested which refers, for example, to the command to be sent from the mobile phone relating, for example, to switching on and off, or more, of the Charging Station.		
T02	IoT Device Discovery Send AR response for Charging Station: *"anchor boxes"*	Through MLaaS the Device is recognized (e.g., Charging Station) by showing a bounding box anchored to the station (e.g. anchor boxes green) on the mobile phone display during video shooting. This anchor box is obtained through the information regarding the geometric coordinates (both normalized and real) provided through Json format which contains this data in the form of a string.	Server EMOT (MLaaS)	Not available yet
T03	IoT Device Discovery Send AR response for Charging Station: *"object detected"*	Through MLaaS the Device is recognized (e.g., Charging Station) also showing the percentage of the object detected during video shooting on the mobile phone display. This % is provided via Json format which contains this data in the form of a string.	Server EMOT(MLaaS)	Not available yet
T04	IoT Device Indexing	Use of the IoT Device Indexing component protected by the IoT Device Access Control. IoT Device	Server ASM (IoT Device Indexing)	Available

	Get a token from Keycloak	Indexing services are accessed by making requests to IoT Device Access Control's Keycloak-protected endpoints. We test the following features: Retrieving a token using Keycloak		
T05	Device Indexing TEST HISTORICAL DATA: requests "*Get all record*".	Use of the Device Indexing component protected by Access Control. Device indexing services are accessed by sending requests to endpoints (protected by Access Control's Keycloak) and We test the following functionality: Retrieving information about the digital twins of the devices and then retrieving the measurements of interest (e.g., %Charge) <u>In this Test in particular, all the records relating to the devices present on the digital twin are requested</u>	Server ASM (Device Index Model)	Available
T06	IoT Device Indexing TEST HISTORICAL DATA: requests "*Get records/ID Device*". **+** <u>Activation in AR mode</u>	Use of the IoT Device Indexing component protected by the IoT Device Access Control. IoT Device Indexing services are accessed by sending requests to endpoints (protected by the IoT Device Access Control's Keycloak) and we test the following functionality. Retrieving information about the digital twins of the devices and then retrieving the measurements of interest (e.g., %Charge) for a specific device identified via QR Code, which will then subsequently activate the control in AR mode by displaying "boxes" on the mobile phone display containing this information which are superimposed on the real image of the charging station during the framing of the station itself In this Test in particular, all the records relating to the Device ID present on the digital twin are requested.	Server ASM (Device Index Model)	Not available yet

T07	Device Indexing TEST HISTORICAL DATA: requests *"Get last n records/ID Device"*	Use of the IoT Device Indexing component protected by the IoT Device Access Control. IoT Device indexing services are accessed by sending requests to endpoints (protected by the IoT Device Access Control's Keycloak) and we test the following functionality. Retrieving information about the digital twins of the devices and then retrieving the measurements of interest (e.g., %Charge) for a specific device identified via QR Code. In this Test in particular, only the last N records relating to the Device ID present on the digital twin are requested.	Server ASM (IoT Device Indexing)	Available

The execution timeline of the aforementioned tests, divided into multiple phases, is detailed in Table 55.

Table 55: AR-based charging management tests execution timeline.

N. Test	Phase	Estimated start date	Estimated end date	Notes
T01 (includes: T07+T06+T05+ T04+T03+T02)	Macro General Test including IoT Device Discovery & IoT Device Indexing + "Actuate Command"	M31	M34	
T02	IoT Device Discovery Send AR response for Charging Station: *"anchor boxes"*	M31	M34	-
T03	IoT Device Discovery Send AR response for Charging Station: *"object detected"*	M31	M34	-
T04	IoT Device Indexing	M30	M31	-

	Get a token from Keycloak			
T05	IoT Device Indexing TEST HISTORICAL DATA: requests "*Get all record*".	M30	M31	-
T06	IoT Device Indexing TEST HISTORICAL DATA: requests "*Get records/ID Device*". **+** activation in AR mode	M31	M34	-
T07	IoT Device Indexing TEST HISTORICAL DATA: requests "*Get last n records/ID Device*"	M30	M31	-

Following it has been reported some screenshots to show part of the results tested as described above.

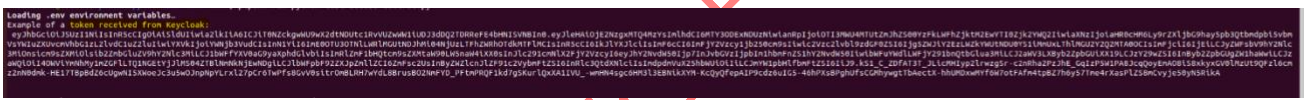


Figure 108: IoT Device Indexing - Get a token from Keycloak.

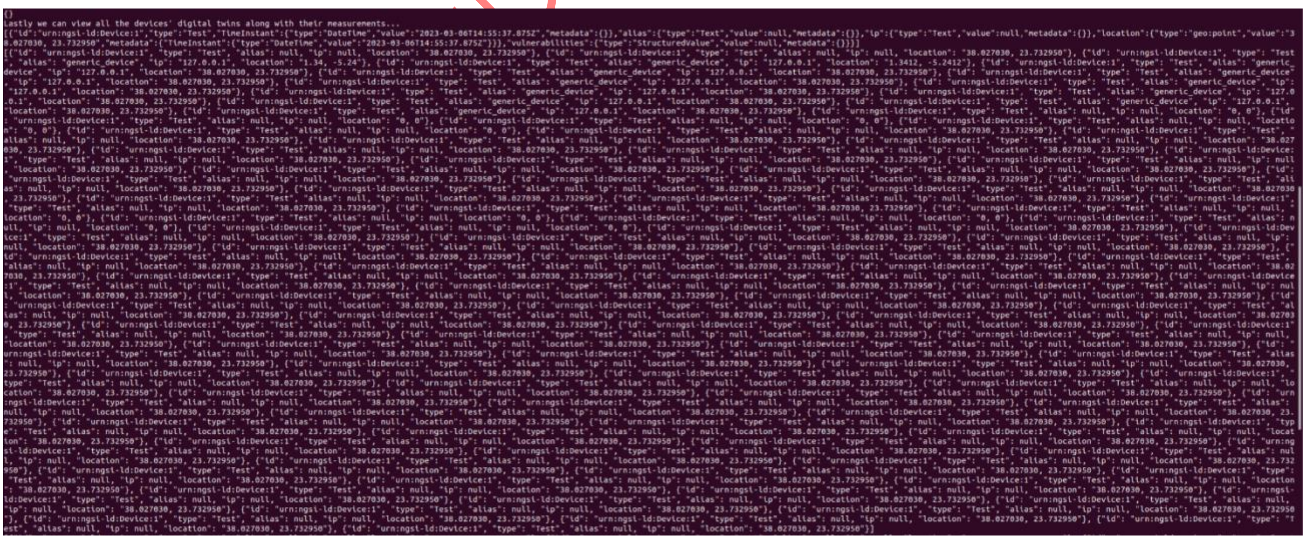


Figure 109: IOT Device Indexing - TEST HISTORICAL DATA: requests "*Get all record*".

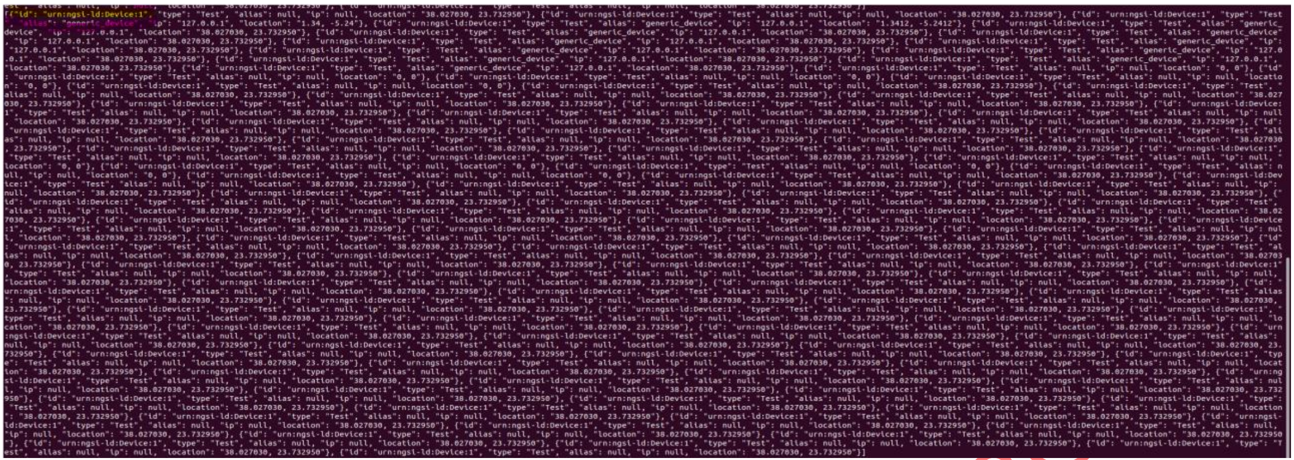


Figure 110: IoT Device Indexing - TEST HISTORICAL DATA: requests "*Get records/ID Device*".

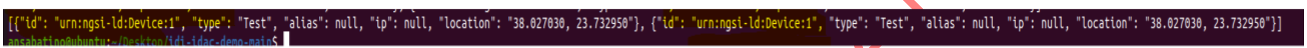


Figure 111: IoT Device Indexing - TEST HISTORICAL DATA: requests "*Get last n records/ID Device*".

Furthermore, the IoT Device Access Control module has been installed to access UC10 resources. It is responsible for the protection of services and the authentication and authorization of users. It employs the Kong Gateway API, an open-source project that enables the use of a variety of security methods like oauth and jwt tokens. An important feature is the ability to create custom plugins to fit the needs of any application. Within the scope of IoT-NGIN, the following custom plugins have been implemented:

- Proximity: A plugin that enables ambient-intelligence based protection. It compares the distance between the device of a user making a request and the requested device and if it is lower than a configurable threshold it allows access.
- Keycloak: A plugin that enables Keycloak authentication for both public and private clients.
- SSI: A plugin that enables authentication bases on the Privacy Preserving Self Sovereing Identities component.

The module also uses the Konga GUI, an interface for Kong that allows quick and easy management of the Kong services and plugins.

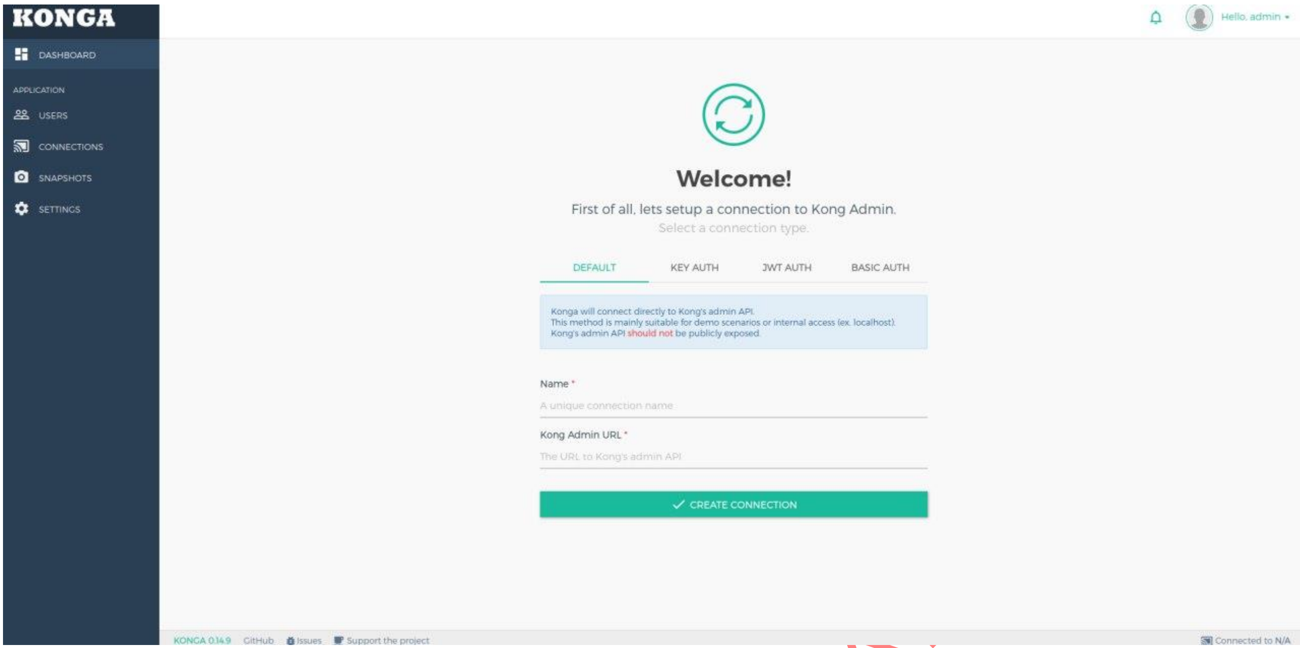


Figure 112: Konga login.

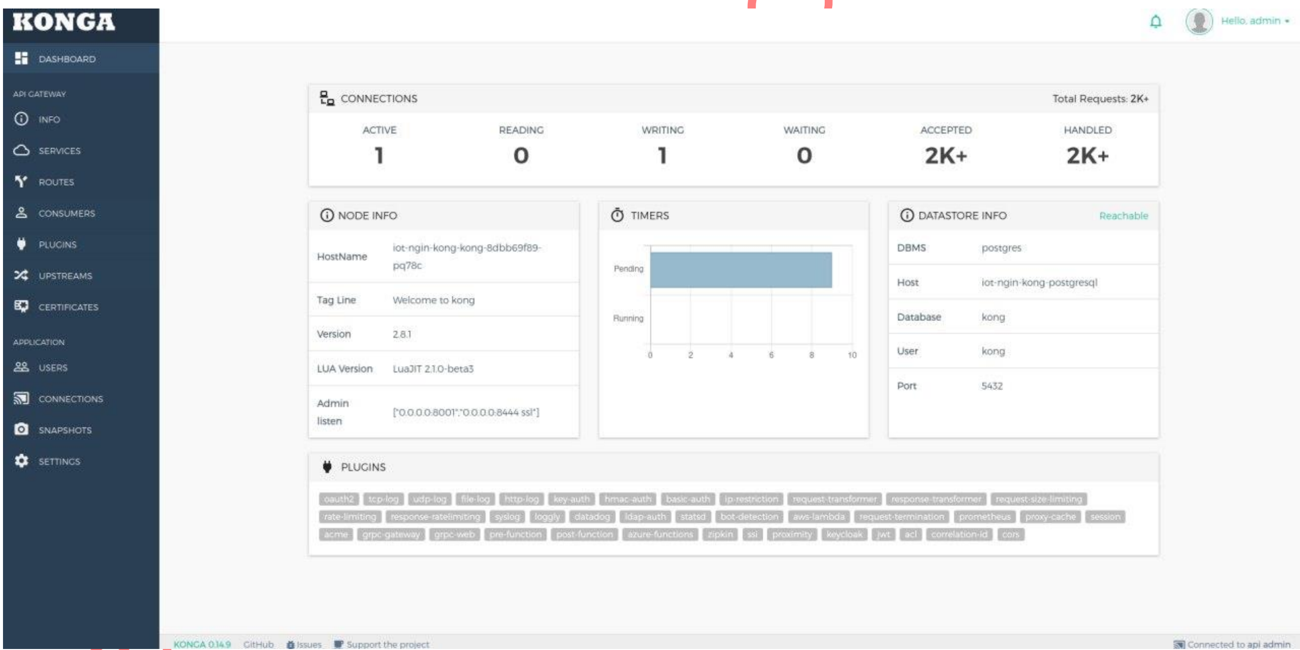


Figure 113: Konga dashboard.

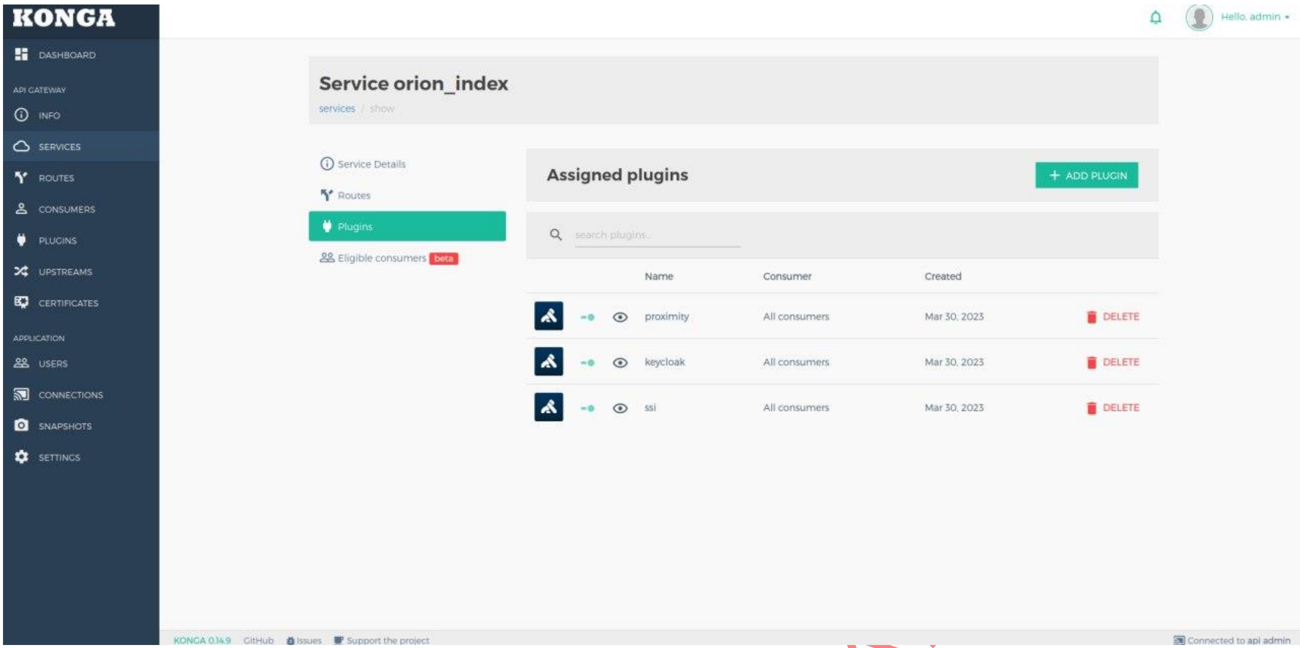


Figure 114: Kong Gateway API plugins.

Finally, charging station APIs for remote management have been implemented to be integrated in AR tool; in particular, three API have been implemented:

- Charging session remote start;
- Charging session remote stop;
- Charging station remote power output setup.

```
fildev@backdev
1 ❯ wget --post-data="tower_id=18&plug_number=2&user_auth=***" https://emotion-projects.eu/api/actions/start-charge/
--2023-03-29 11:32:39--  https://emotion-projects.eu/api/actions/start-charge/
Loaded CA certificate '/etc/ssl/certs/ca-certificates.crt'
Resolving emotion-projects.eu (emotion-projects.eu)... 82.165.124.120
Connecting to emotion-projects.eu (emotion-projects.eu)|82.165.124.120|:443... connected.
HTTP request sent, awaiting response... 204 No Content
2023-03-29 11:32:39 (0.00 B/s) - ‘index.html’ saved [0]
```

Figure 115: EMOT charging session remote start API.

```
fildev@backdev
❯ wget --post-data="tower_id=18&transaction_id=65c0ad74-a91d-4367-ad86-f29f9ca4aeba" https://emotion-projects.eu/api/actions/stop-charge/
--2023-03-29 11:35:20--  https://emotion-projects.eu/api/actions/stop-charge/
Loaded CA certificate '/etc/ssl/certs/ca-certificates.crt'
Resolving emotion-projects.eu (emotion-projects.eu)... 82.165.124.120
Connecting to emotion-projects.eu (emotion-projects.eu)|82.165.124.120|:443... connected.
HTTP request sent, awaiting response... 204 No Content
2023-03-29 11:35:20 (0.00 B/s) - ‘index.html’ saved [0]
```

Figure 116: EMOT charging session remote stop API.

```
fildev@backdev
❯ wget --post-data="tower_id=18&hw_type=dc&power=200" https://emotion-projects.eu/api/actions/set-power/
--2023-03-29 11:59:21--  https://emotion-projects.eu/api/actions/set-power/
Loaded CA certificate '/etc/ssl/certs/ca-certificates.crt'
Resolving emotion-projects.eu (emotion-projects.eu)... 82.165.124.120
Connecting to emotion-projects.eu (emotion-projects.eu)|82.165.124.120|:443... connected.
HTTP request sent, awaiting response... 204 No Content
2023-03-29 11:59:21 (0.00 B/s) - ‘index.html’ saved [0]
```

Figure 117: EMOT charging station remote power output setup.

4 Cross-living lab validation

Cross-living Lab validation has been shifted towards IoT-NGIN architecture validation under the Living Lab umbrella. This architecture validation approach has already been instantiated within the WP6.

In order to justify the ability of IoT-NGIN architecture Figure 118 to generally fit all use cases, an analysis of the architecture enablers alignment with different Living Labs use cases has been done.

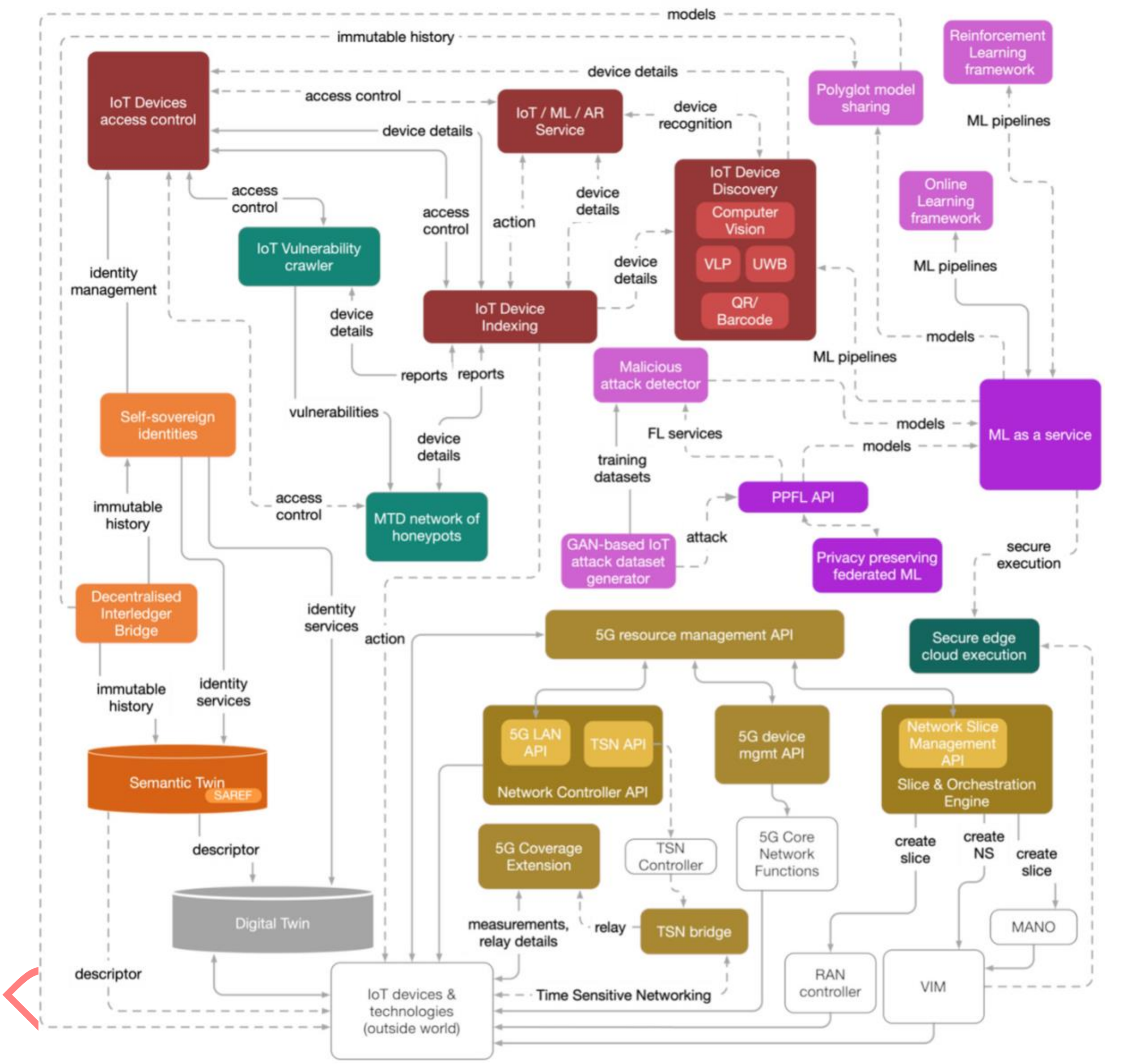


Figure 118: The IoT-NGIN logical architecture.

4.1 IoT-NGIN Architecture components

The IoT-NGIN project provides multiple components to improve and deploy solutions that are utilised in the architecture instantiations of different Living Labs use cases (UCs). Table 56 below shows a summary of each use case with the different IoT-NGIN technologies. It lists the different IoT-NGIN technologies grouped by their corresponding WP and which Living Labs and use cases they are involved in.

Table 56: Summary of the alignment of IoT-NGIN technologies with the different Living Labs.

IoT-NGIN Technology	Smart Cities			Smart Agriculture		Industry 4.0			Smart Energy	
	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10
WP2 – Enhancing IoT Underlying Technology										
Use of 5G networks in trial sites	✓	✓	✓	*Public 5G not available yet at trial sites*			✓	✓	✓	✓
Secure edge-cloud execution framework				✓	✓					
WP3 – Enhancing IoT Intelligence										
MLaaS framework	✓	✓	✓	✓	✓	✓		✓	✓	✓
Deep Learning, Reinforcement Learning & Transfer Learning				✓	✓				✓	✓
Privacy-preserving Federated Machine Learning	✓	✓	✓	✓						
Polyglot Model Sharing component	✓	✓	✓	✓				✓		
WP4 – Enhancing IoT Tactile & Contextual Sensing/Actuating										
IoT device discovery (Computer Vision)	✓	✓		✓		✓		✓		✓
IoT device discovery - Non-visual (RFID, QR, UWB)				✓		✓	✓	✓		
IoT device indexing	✓	✓		✓	✓	✓	✓	✓		✓

IoT-NGIN Technology	Smart Cities			Smart Agriculture		Industry 4.0			Smart Energy	
	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10
IoT device access control	✓	✓		✓	✓					✓
IoT AR toolkit				✓		✓	✓			✓
WP5 – Enhancing IoT Cybersecurity & Data Privacy										
Generative Adversarial Networks (GAN) based dataset generation				✓					✓	✓
Malicious Attack Detector (MAD)				✓					✓	✓
IoT vulnerability crawler	✓	✓		✓					✓	✓
Moving Target Defences (MTD) Network of Honeypots				✓						
Decentralized Interledger Bridge	✓	✓						✓		✓
Privacy preserving Self-Sovereign Identities (SSIs)	✓	✓	✓						✓	✓
Semantic Twins	✓	✓	✓					✓	✓	

4.2 Analysis

The analysis focuses on examining the IoT-NGIN architecture from four angles: functional layers, components dependency, reusability and adaptability.

Functional layers

The IoT-NGIN architecture presented in Figure 118, as a set of functional layers, namely the Federated Communications layer, the Micro-services and Virtual Network Functions (VNFs), the Federated Data Sovereignty, the Federation of Big Data Analytics & ML and the Human-Centred Augmented Reality Tactile IoT. Each functional layer is supported by a set of components provided by different technical WPs.

Components for 5G enhancements indicate comprehensively the enhancements and 5G capabilities exposed by the relevant APIs, as well as the way they manage 5G resources. The "IoT Device Discovery" component developed in IoT-NGIN, comes in four variants, including the Computer Vision, Visual Light Positioning (VLP), Ultra-Wide Band (UWB) and Code Scanning (QR/barcode) variants.

The "Deep Learning, Reinforcement Learning & Transfer Learning framework" has been split into "Reinforcement Learning framework" and "Online Learning framework", which both connect to the MLaaS framework.

The Privacy-Preserving Federated Learning (PPFL) API has been added as an API to consume federated learning services across the different FL frameworks considered in the project. The term "Semantic Twin" has been adopted to align with the definition and scope of the component developed, instead of the term "Meta-Level Digital Twin".

Components dependency

In terms of UCs dependency to the IoT-NGIN components, the table above and the description of the architecture components of each UC indicates that at least three IoT-NGIN components are used in all UCs. We can differentiate three level of usage. At the top level the UC1, UC2, UC4 and UC10 rely on 10 or more enablers. Regarding the second level UC3, UC5, UC6 use between 5 or and 7 enablers. Finally, UC7 uses only three enablers.

Components reusability

Regarding reusability, all IoT-NGIN components are implemented by at least three UCs. Some components have higher usage rate such as MLaaS framework that it is implemented in 9 out of 10 UCs.

Components adaptability

Concerning the adaptability. The majority IoT-NGIN components are adaptable with no or slight changes to fit the UCs requirements.

IoT-NGIN components provide various solutions that are operational and reusable under all UCs, therefore the IoT-NGIN architecture can be considered as a mature enough alternative solution for cross living lab validation.

5 Conclusions

This deliverable included a summary of the progress made in the IoT-NGIN Living Labs, based on the trial site setup, equipment procurement, alignment with IoT-NGIN technologies, definition of testing and validation processing given on the D7.2 – *IoT-NGIN Living Labs use cases initial results* deliverable, including any other intermediate results obtained so far.

After finalizing the intermediate stage of trial set-up and definition, next steps will focus on finishing the pilot activities in the different Living Labs. Given the wide variety of Living Labs in IoT-NGIN, each trial will continue its own execution timeline and stages, as defined in this deliverable. Nonetheless, final updates about the results of all Living Labs are expected by M36 of the project and will be summarized in *D7.4 – IoT-NGIN Living Labs use cases Assessment and Replication guidelines.*

No further updates about the DMP are expected at this point. All relevant information about the trial datasets, such as technical characteristics, data controller, access levels and other FAIR data considerations, has already been included in the D7.2 – *IoT-NGIN Living Labs use cases initial results* deliverable. Should the need arise to make any addition or correction to the DMP at the end of the project, these updates will be included in the deliverable *D7.4 – IoT-NGIN Living Labs use cases Assessment and Replication guidelines*, due at M36.

References

[1] IoT-NGIN, "D7.2 – Trial site set-up, initial results and DMP update," H2020 - 957246 - IoT-NGIN Deliverable Report, 2022.

[2] Synelixis, "SynField," Synelixis, [Online]. Available: https://www.synfield.gr/. [Accessed 2023].

[3] IoT-NGIN, "D1.1 - Definition analysis of use cases and GDPR Compliance," H2020 957246 - IoT-NGIN Deliverable Report, 2021.

[4] benc-uk, "KubeView," [Online]. Available: https://kubeview.benco.io. [Accessed 2023].

[5] IoT-NGIN, "D6.2 - Integrated IoT-NGIN platform & laboratory testing results," H2020 - 957246 - IoT-NGIN Deliverable Report, 2022.

[6] M. Alessandrini, R. Rivera, C. Fuentes, L. Falaschetti, D. Pau, V. Tomaselli and C. Turchetti, "A grapevine leaves dataset for early detection and classification of esca disease in vineyards through machine learning," *Data in Brief*, p. 106809, 2021.

[7] "Kaggle," Kaggle, 25 12 2020. [Online]. Available: https://www.kaggle.com/code/nageshsingh/nature-image-classification. [Accessed 15 3 2023].

[8] Open Information Security Foundation, "Suricata," [Online]. Available: https://suricata.io/. [Accessed 2023].

[9] jasonish, "Suricata Docker Image," Github, [Online]. Available: https://github.com/jasonish/docker-suricata. [Accessed 2023].

[10] ROS, "Robot Operating System," [Online]. Available: https://www.ros.org. [Accessed 2022].

[11] Open Source Robotics Foundation, "Open Robotics," [Online]. Available: https://www.openrobotics.org/. [Accessed 2023].

[12] R. R. Shamshiri, I. A. Hameed, M. Karkee and C. Weltzien, "Robotic Harvesting of Fruiting Vegetables, A Simulation Approach in V-REP, ROS and MATLAB," in *Automation in Agriculture - Securing Food Supplies for Future Generations*, INTECH, 2018.

[13] StereoLabs, "Getting Started with ROS and ZED," 2021. [Online]. Available: https://www.stereolabs.com/docs/ros/. [Accessed 2023].

[14] ROS.org, "rviz," 2018. [Online]. Available: http://wiki.ros.org/rviz. [Accessed 2023].

[15] Open Robotics, "Gazebo," [Online]. Available: https://gazebosim.org/home. [Accessed 2023].

[16] G. Jocher , A. Chaurasia, A. Stoken, . J. Borovec, N. Y. Kwon, K. Michael, T. J. Fang, i. L. Z. Yifu, C. Wong, A. V. D. Montes, Z. Wang, . C. Fati, J. Nadar, L. U. V. Sonck, t. y. P. Skalski, A. Hogan, D. Nair, M. Strobel and M. Jain, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime," *Zenodo,* 2022.

[17] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision,* vol. 111, pp. 98-136, 2015.

[18] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick and P. Dollár, "Microsoft COCO: Common Objects in Context," in *ECCV: European Conference on Computer Vision*, Zurich, 2014.

[19] IoT-NGIN, "D3.3 - Enhanced IoT federated deep learning/ reinforcement ML," H2020 - 957246 - IoT-NGIN Deliverable Report, 2022.

[20] IoT-NGIN, "D3.4 - ML models sharing and Transfer learning implementation," H2020 - 957246 - IoT-NGIN Deliverable Report, 2023.

[21] H. R. Roth, Y. Cheng, Y. Wen, I. Yang, Z. Xu, Y.-T. Hsieh, K. Kersten, A. Harouni, C. Zhao, K. Lu, Z. Zhang, W. Li, A. Myronenko, D. Yang, S. Yang, N. Rieke, A. Quraini and C. Chen, "NVIDIA FLARE: Federated Learning from Simulation to Real-World," in *International Workshop on Federated Learning, NeurIPS 2022*, New Orleans, USA, 2022.

[22] IoT-NGIN, "D4.3 - Enhancing IoT Tactile & Contextual Sensing/Actuating," H2020 - 957246 - IoT-NGIN Deliverable Report, 2022.

[23] ROS.org, "rosserial_python," [Online]. Available: http://wiki.ros.org/rosserial_python. [Accessed 2023].

[24] Electric Sheep Fencing, "pfSense," [Online]. Available: https://www.pfsense.org/. [Accessed 2023].

[25] OISF, "Suricata," [Online]. Available: https://suricata.io.

[26] E. Commission. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2018.328.01.0082.01.ENG&toc=OJ:L:2018:328:TOC.

[27] SOFIE, "Secure Open Federation for Internet Everywhere," 2018-2020. [Online]. Available: https://www.sofie-iot.eu/.

[28] [Online]. Available: https://github.com/SOFIE-project/Marketplace.

[29] Food and Agriculture Organization, "Climate change fans spread of pests and threatens plants and crops, new FAO study," 2021. [Online]. Available: https://www.fao.org/news/story/en/item/1402920/icode/.

[30] Food and Agriculture Organization of the United Nations (FAO), " COVID-19 Series / Improving the Storability and Availability of Pulses by Reducing Post-Harvest Losses,"

[Online]. Available: http://www.fao.org/food-loss-reduction/news/detail/en/c/1271760/. [Accessed 2020].

[31] European Seed, "Vegetable Growers Looking to Accelerate Switch to Mechanical Harvesting Due to Coronavirus Crisis," 2020. [Online]. Available: https://european-seed.com/2020/06/vegetable-growers-looking-to-accelerate-switch-to-mechanical-harvesting-due-to-coronavirus-crisis/.

[32] AgriLinks, "Climate-Smart Agriculture Could Offset COVID-19's Impact on Bangladeshi Farmers," 2020. [Online]. Available: https://www.agrilinks.org/post/climate-smart-agriculture-could-offset-covid-19s-impact-bangladeshi-farmers.

[33] IoT-NGIN, "D7.1 - Data Management Plan (DMP)," H2020 957246 - IoT-NGIN Deliverable Report, 2021.

[34] F. Domingues and J. S. d. Aguila, "The cost of grape mechanical harvesting is more economical than the manual harvest?," in *39th World Congress of Vine and Wine*, 2016.