This project has received funding from the European's Union Horizon 2020 research innovation programme under Grant Agreement No. 957258

Architecture for Scalable, Self-human-centric, Intelligent, Secure, and Tactile next generation IoT

D6.8 Final Integration and Support Report

Deliverable No.	D6.8	**Due Date**	31-OCT-2023
Type	Report	**Dissemination Level**	Public
Version	1.0	**WP**	WP6
Description	Final report of WP6 activities, comprising the final activities related to testing, integration, packaging, releasing and documentation.		

Copyright

The ASSIST-IoT consortium consists of the following 15 partners:

UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Spain
PRODEVELOP S.L.	Spain
SYSTEMS RESEARCH INSTITUTE POLISH ACADEMY OF SCIENCES IBS PAN	Poland
ETHNIKO KENTRO EREVNAS KAI TECHNOLOGIKIS ANAPTYXIS	Greece
TERMINAL LINK SAS	France
INFOLYSIS P.C.	Greece
CENTRALNY INSTYUT OCHRONY PRACY	Poland
MOSTOSTAL WARSZAWA S.A.	Poland
NEWAYS TECHNOLOGIES BV	Netherlands
INSTITUTE OF COMMUNICATION AND COMPUTER SYSTEMS	Greece
KONECRANES FINLAND OY	Finland
FORD-WERKE GMBH	Germany
GRUPO S 21SEC GESTION SA	Spain
TWOTRONIC GMBH	Germany
ORANGE POLSKA SPOLKA AKCYJNA	Poland

Disclaimer

Authors

Name	Partner	e-mail
Alejandro Fornés	P01 UPV	alforlea@upv.es
Juan Gascón	P01 UPV	juagasre@upv.es
Rafael Vañó	P01 UPV	ravagar2@upv.es
Raúl Reinosa	P01 UPV	rreisim@upv.es
Eduardo Garro	P02 PRODEVELOP	egarro@prodevelop.es
Piotr Sowiński	P03 SRIPAS	psowinski@ibspan.waw.pl
Pawel Szmeja	P03 SRIPAS	pawel.szmeja@ibspan.waw.pl
Karolina Bogacka	P03 SRIPAS	karolina.bogacka@ibspan.waw.pl
Evripidis Tzionas	P04 CERTH	tzionasev@iti.gr
Konstantinos Fragkos	P06 INFOLYSIS	cfragkos@infolysis.gr
Nikolaos Vrionis	P06 INFOLYSIS	nvrionis@infolysis.gr
Monika Kobus	P07 CIOP-PIB	mokob@ciop.lodz.pl
Johan Schabbink	P09 NEWAYS	johan.schabbink@newayselectronics.com
Nasia Balakera	P10 ICCS	nasia.balakera@iccs.gr
Fotios Konstantinidis	P10 ICCS	fotios.konstadntinidis@iccs.gr
Tom Papaioannou	P10 ICCS	thomas.papaioannou@iccs.gr
Konstantinos Routsis	P10 ICCS	konstantinos.routsis@iccs.gr
Rafael Borne	P13 S21SEC	rborne@s21sec.com
Zbigniew Kopertowski	P15 OPL	zbigniew.kopertowski@orange.com

History

Date	Version	Change
18-Jul-2023	0.1	ToC and task assignments
12-Sep-2023	0.2	Beginning of first round of contributions
2-Oct-2023	0.3	Beginning of second round of contributions
17-Oct-2023	0.4	End of second round of contributions
26-Oct-2023	0.9	Version for internal review
31-Oct-2023	1.0	Version released to EC

Key Data

Keywords	Software, integration, release, Helm, packaging, GitLab, testing,
Lead Editor	P04 CERTH – Evripidis Tzionas
Internal Reviewer(s)	P01 UPV – Alejandro Fornés, P13 S21SEC – Rafael Borné

Executive Summary

This deliverable is written in the framework of WP6 – Testing, Integration and Support of **ASSIST-IoT** project under Grant Agreement No. 957258. The deliverable is the last iteration for testing and integration, packaging and releasing and technical support and documentation within the ASSIST-IoT project. In order to understand the development and progress of the methodology and implementation, the contemplation of previous deliverables is mandatory.

The context of this deliverable is precise, and it concludes the final stages of software testing. The focus is on the validation of the course of the applications delivered to the pilots (end-to-end), the verification of the requirements (acceptance) and the quantification of performance under different conditions (performance).

It documents the enablers' final packaging status, GitHub, ArtifactHub and DockerHub public repositories structure and setup, where all the code, charts and images are stored respectively. Additionally, it covers software licensing schemes, and addresses third-party IPR compliance.

It concludes every enabler's documentation and support, pointing to ASSIST-IoT ReadTheDocs wiki and to the aforementioned repositories, thus serving as both a developer and end user documentation repository. Finally, it incorporates the second round Open Caller's questionnaires, their results and feedback.

Table of contents

List of tables

List of figures

List of acronyms

Acronym	Explanation
API	Application Programming Interface
BSD	Berkeley Source Distribution (license)
CI/CD	Continuous Integration and Continuous Delivery/Deployment
CRD	Custom Resource Definition
DLT	Distributed Ledger Technology
GNU GPL	GNU General Public License (license)
GWEN	Gateway and Edge Node
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
K8s	Kubernetes
LTSE	Long-Term Storage Enabler
MIT	Massachusetts Institute of Technology (license)
MR	Mixed Reality
NGIoT	Next-Generation Internet of Things
OS	Operating System
OSM	Open Source Mano
PVC	Persistent Volume Claim
SAST	Static Application Security Testing

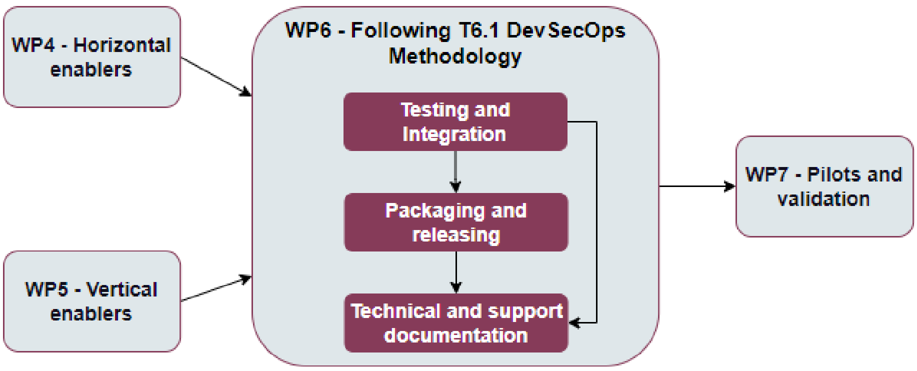
SCTP	Stream Control Transmission Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network
YAML	YAML Ain't Markup Language
IPR	Intellectual Property Rights
OSH	Occupational Safety and Health

1. About this document

This deliverable corresponds to the final actions of WP6 as the project reaches its final stage. The fact that all the three deliverable series that ended on M30 could not be postponed until M36 signalled the conceptualisation of this deliverable, thus enabling to finalise all the procedures in the context of this work package, without a large timespan since the last report. The scope of this deliverable is to document the concluding actions for testing, outline the final structure of the public repositories and define the licensing schemes. Additionally, it is utilised to host all the information summarising the software documentation of the project, alongside the second round of the Open Callers questionnaire and feedback.

Because of the extension of the project, this final deliverable concludes three deliverable series that covered all the aspects of testing & integration, packaging & releasing and technical support & documentation.

1.1. Deliverable context

Keywords	Lead Editor
Objectives	**O1:** This deliverable contributes to the implementation of an NGIoT architecture, by refining the release planning, licensing strategy and leveraged tools for packaging and storing the developed artifacts (in the form of enablers). **O2 to O5:** All the specific implementations associated to these objectives are delivered as packages defined in this document. **O6:** The enablers' releases will feed pilots, which are the places where these are being validated. Since three deliverable series close with this combined document the objectives expand to: • Delivering complete documentation of the final phases of software testing and integration activities. • Concluding the packaging strategy, conventions and tools, while presenting the polished registries and licensing strategy. • Developing and releasing supporting documentation for both 3rd parties participating in Open Calls and Stakeholders, together with the Open Source Community publishing
Work plan	The tasks associated to this deliverable deals with all packaging and releasing aspects of the enablers developed in WP4 & WP5 (T6.3). They offer a complete set of guidelines and methodology of testing and integration procedures, emphasising on the final phases closer to leading to the end product (T6.2) and finally, a documentation to assist both technical personnel as well as third parties to integrate, use, or extend the technical capabilities of ASSIST-IoT (T6.4). 
Milestones	This deliverable directly contributes to MS7 - *Integrated solution* (initially planned for M30) and to the MS6 - *Software structure finished* (M24). The deliverable is the final report of

	testing methods and integration, packaging status and release plan, technical support and documentation, as well as Open Caller's feedback.
Deliverables	All WP6 tasks resulted in this current deliverable. Starting with the Testing and Integration Plan (D6.3 [1.]), other concurrent deliverables within WP6 complement this document. These include the release and distribution plan (D6.7 [2.]) and documentation (D6.6 [3.]). These three deliverables themselves are parts of their series that document the evolution of WP6 tasks throughout the project. The last deliverable of this WP (D6.8) contains outcomes from all the tasks, although it is essential to engage the entire deliverable series for a concrete understanding of these results.

1.2. The rationale behind the structure

This deliverable serves as a consolidation of three key deliverable series: the Testing and Integration Plan, Packaging and Releasing, and Technical Support and Documentation. Hence, Section 2 aims in providing a comprehensive overview of the strategy and the results of the final stages of testing and integration, with due consideration to the previous stages already covered in D6.3 [1.]. Section 3 encompasses the final packaging and release plan. It includes details regarding the structure of GitHub, Docker Hub and Artifact Hub repositories, as well as insights into the software licensing approach. Section 4 is dedicated to the exhaustive documentation of each of the enablers, accompanied by the open callers' questionnaires and feedback, providing a comprehensive view of ASSIST-IoT progress. Finally, Section 5 offers conclusions, summarising the expected outcomes in the final deliverable of WP6.

1.3. Outcomes of the deliverable

The main outcome of this deliverable is the documentation, summarising the finalisation of the activities of WP6. The work progress is interconnected with the preceding deliverable series, and due to the project's extension, a final report is necessary to compile all the work aspects that have been developed. D6.3 [1.], D6.6 [3.] and D6.7 [2.] serve as the foundation of the outcomes presented in this deliverable:

The first section provides a detailed account of the first three testing phases undertaken in the project. While unit, functional and integration testing have been already conducted, end-to-end, acceptance and performance testing take over with explanation of their methodologies and results.

Section 3 reports the enabler's packaging status and emphasises on the software licensing schemes, in conjunction with third party IPR compliance. The latter part of the section focuses on the establishment of GitHub, ArtifactHub and DockerHub repositories, intended for the dissemination of ASSIST-IoT's software solution.

Section 4 serves as a repository of final supporting documentation tailored for developers' technical utilisation, in conjunction with Stakeholders, 3d party Open Callers and Open Source Community publishing. This section also includes the conclusive Open Callers questionnaire along with its results and feedback, which will inform future improvements.

1.4. Lessons learnt

Over the past six months, reflecting on the latest achievements in WP6, the project has continued to gain valuable lessons that persistently apply to the current state. Building upon the insights shared earlier, several notable understandings have emerged:

- Although requirements are conceptualised early in each project, their significance becomes pronounced during acceptance testing. Evaluating them in this context is an accurate and equitable approach. It is crucial to foresee and predict what should be implemented later, finding ways to evaluate them effectively.

- Some technical KPIs, designed early and refined as the project progresses, must be structured for measurability. Incorporating KPIs into performance testing poses challenges not only for the initial design but also for quantifying and measuring performance.
- In the scope of the project, packaging is a simple process; however, to unlock its true potential, configurability has to be ensured. This entails that relevant environment values from containers, bounded to the code, should be passed to them via Helm values, and therefore charts have to be defined taking it into account.

1.5. Deviation and corrective actions

- Although most enablers had packaged versions back in M30 (April 2023), additional tests outside laboratory conditions (in pilots) showed that there was room for improvement, especially in terms of configuration possibilities and reliability of data persistency. New charts have been made available in the lasts months.
- Some documentation was not properly reported in the project's ReadTheDocs, but in GitLab's readmes. That caused confusion when implementing them in some pilots. Now, all enablers have proper documentation in ReadTheDocs, without precluding the possibility of having extensive readme files.
- The initial approach to reporting acceptance testing involved documenting all requirements, both pilot and common, along with their validation procedures in this deliverable. However, due to variations in the maturity of the pilots and the ongoing documentation of some validation procedures, it was determined that consolidating them in D8.3 would lead to more comprehensive and consolidated reporting.

1.6. Version-specific notes

In comparison to the previous deliverables of WP6, this deliverable is a combination of three series, so it contains updates for all of them, making it difficult to dive deep into details about each section. Thus, it encompasses the updates of (i) the final stages of testing and integration, (ii) the final setup of the repositories for enablers, including software licensing and third party IPR compliance and (iii) the final documentation and support for every ASSIST-IoT enabler, in addition to questionnaire results and Open Callers' feedback.

2. Testing and integration

In the previous iterations of the deliverable, the DevSecOps methodology, infrastructure and integration tools were introduced. ASSIST-IoT hosts all the state-of-the-art integration tools, with GitLab being the main protagonist and GitLab CI/CD, GitLab Runner, Helm, and container registries utilised throughout the project. The testing environment infrastructure includes physical machines, VMs, VPN enabler and K8s clusters.

The current section's primary objective is to present the latest updates regarding the testing methodology and its implementation. For a holistic understanding of the infrastructure and tools harnessed in integration, referencing previous versions of the deliverables is necessary. This enables a traceable view of the methodology's evolution over the project's timespan.

GitLab's functions are outlined in D6.3 [1.]. GitLab CI/CD is elaborated in section 2.1.1, with an example `.gitlab-ci.yml` file accompanied by detailed guidelines on how to use it. In section 2.1.2, GitLab Runner is detailed, offering exhaustive instructions on installation, startup, and the anticipated outcomes. Sections 2.1.3 and 2.1.4 encompass all the information regarding the process of pushing Helm charts and Docker images, as well as using them, within Helm and Container registry, respectively. Finally, the testing/integration environment is unfolded in section 2.2 explaining its structure and functionality.

Regarding the testing phases, D6.3 contains extensive documentation about the methodology followed in each phase. It is worth highlighting while unit testing guidelines were proposed, the testing results were not reported for the sake of time consumption. Functional and integration testing have been carried out and the reported results can be found in D6.3's section 4.1 and 4.2 respectively. The primary focus of this deliverable is on the subsequent phases of testing and providing a comprehensive coverage of the results.

2.1. Functional testing

Functional testing is a critical aspect of the testing process that assesses the functionality of enablers. Moving forward from unit testing, this phase of testing evaluates whether the system behaves as expected when provided with specific inputs, ensuring it adheres to defined enabler requirements. In the context of the project, D6.2 [4.] previously hosted a documentation of the functional testing potential efforts. In that deliverable, the tests to be executed were outlined, while D6.3 [1.] provided a refined list of functional tests, in addition to a detailed account of the testing procedures and outcomes, covering both manual and automated tests.

In continuation of the information provided earlier, this specific deliverable primarily focuses on presenting the latter phases of testing, hence the previous content will not be repeated. Nevertheless, it is important to note that comprehensive details pertaining to the testing approach, the tools employed, pre-test conditions, and the sequence of tests can be found in D6.3, specifically in section 4.1. This section of D6.3 provides an in-depth account of the methodology and procedures related to functional testing.

2.2. Integration testing

The integration testing methodology and approach within ASSIST-IoT were initially outlined in D6.2 [4.]. In D6.3 a decision was made to adopt a different approach to integration testing, specifically focusing on testing the integrations between enablers themselves. The primary objective became testing the connections between enablers, without delving into the intricate details of their internal processes, which had already undergone testing during the unit and functional testing phases.

This approach marks the initial phase of integrating various components to construct complete pilot trials during the subsequent end-to-end testing phase. The table provided below offers an overview of the current status of integrations that have been implemented within the project. Partners who have been actively involved in implementing these integrations are highlighted in blue.

Table 1: Integration progress of ASSIST-IoT enablers

#	Enablers Involved	Integration Description	Partner Responsible & Involved	Pilots Involved	Current status
1	**EDBE – LTSE**	The integration involves the information (camera position, licence plate number etc.) of the pictures that arrive from the scanners, which have to be transferred through EDBE. The information is sliced into separate attributes and then stored to the LTSE.	CERTH - TT	P3b	Completed
2	**Monitoring & Notifying – ALL DLT enablers**	Integration of Kafka and DLT enablers. Firstly, Logging and Auditing in order to store critical event notifications from IoT devices. Secondly with Integrity Verification to store hashes of information in order to ensure that it remains intact. Third with Broker Service to monitor the status of edge devices and gateways.	CERTH	P3b	Completed
3	**OpenAPI - IdM**	Configuring Kong API Gateway to use Open ID Connect (OIDC) plugin to integrate with Keycloak IdM in order to secure exposed endpoints. Also using Keycloak OIDC to connect to the openAPI Portal.	CERTH - S21SEC	ALL	Completed
4	**Semantic Annotation enabler – Semantic Repository enabler**	Downloading RML files from the Semantic Repository. The Semantic Annotation enabler needs these files to annotate incoming data streams. The integration will be used in Pilot 2 to integrated data from various sources.	SRIPAS	P2	Completed
5	**Semantic Annotation enabler – EDBE**	Integration of the MQTT protocol with the Semantic Annotation enabler. The Semantic Annotation enabler is able to produce and consume data using the MQTT protocol. The integration is needed for Pilot 2 to ensure interoperability with other enablers and components.	SRIPAS - ICCS	P2	Completed
6	**Semantic Translation enabler – EDBE**	In addition to the existing Apache Kafka support, Semantic Translation enabler will offer integration of the MQTT protocol.	SRIPAS – ICCS	P2	Completed
7	**Location Processing enabler – EDBE**	Integration of the MQTT protocol with the Location Processing enabler. The Location Processing enabler is able to produce and consume data using the MQTT protocol. The integration is needed for Pilot 2 to ensure interoperability with other enablers and components.	SRIPAS – ICCS	P2	Completed
8	**Location Processing enabler – Location Tracking enabler**	Configuration in the Location Tracking enabler was prepared for ingesting location data from the location tags. A common schema for data exchanged between the enablers was established. The integration is needed for location-tracking related use cases in Pilot 2.	SRIPAS – NEWAYS	P2	Completed

#	Enablers Involved	Integration Description	Partner Responsible & Involved	Pilots Involved	Current status
9	**MR enabler – Location Processing enabler**	Obtaining worker locations from the Location Processing enabler through MQTT protocols, using the EDBE and displaying those new locations inside the MR enabler. The MR enabler, as soon as it receives the MQTT message, it compares the tags in the message with the workers' database and displays the new location of the workers that are closer to the MR enabler.	SRIPAS – ICCS	P2	Completed
10	**MR enabler – Semantic Repository enabler**	Downloading 3D models and other media files from the Semantic Repository using REST API and displaying them through the MR enabler.	ICCS – SRIPAS	P2	Completed
11	**Smart orchestrator - PUD**	To let the orchestrator decide the optimal place to automatically deploy enablers in the available clusters, different metrics (resource and latency-related) are needed. This required implementing a federated version of the PUD enabler to access such information from a central location, without needing to involve the LTSE nor custom synchronisation mechanisms.	UPV - ICCS	P2	Completed
12	**Smart orchestrator - Manageability enablers**	The Smart orchestrator API is extensive and unfriendly to be directly utilised for administrating a given deployment (i.e., involved clusters, enablers and repositories). Manageability enablers provide interfaces and forms to abstract it, which required integrating its backend with the right orchestrator endpoints, including some aggregation of calls and filtering of the results.	UPV	ALL	Completed
13	**Smart orchestrator - EDB**	ETSI MANO architecture was not thought for massive deployments, in which several clusters (managed under the umbrella of "groups") will deploy the same set of enablers. To cope with this kind of cases, an extended MQTT-based architecture was implemented, in which the orchestrator incorporates a dedicated MQTT client and data model to manage the entire flow. MQTT bridges between the main EDBE and edge instances have been also provisioned. Buffering and timeout strategy under refinement.	UPV	ALL	Completed
14	**SD-WAN & WAN acceleration enablers**	These enablers are naturally working together to implement tunnels among connected network sites. Dedicated code was needed to enable the configuration of the WAN acceleration enabler instances from the SD-WAN controller, which acts as the orchestrator of the entire SD-WAN solution.	UPV	ALL	Completed
15	**VPN enabler - LTSE**	Information about clients provisioned was initially stored in an internal database of the VPN enabler. This information has been moved to the LTSE, and to that end the API of the VPN enabler has been modified to implement the required LTSE endpoints. In this way, less storage systems are needed in a given deployment,	UPV	ALL	Completed

#	Enablers Involved	Integration Description	Partner Responsible & Involved	Pilots Involved	Current status
16	**Manageability enablers - Tactile dashboard**	Manageability enablers are essential to manage a deployment, and therefore it was natural to integrate them with the main user framework of the project. To that end, the development guides of the tactile dashboard enabler were leveraged to generate such interfaces (including backends) and then integrating them in the same solution (VUI + Spring /PUI9 framework), avoiding the need of deploying different web applications independently and optimizing in this way the available resources.	UPV – PRO	ALL	Completed
17	**Traffic classification enabler - Semantic repository enabler**	The traffic classification enabler can generate and needs models to work. At this moment, involved models are stored and consumed in a storage volume of the host containing an instance of the enabler. The idea is to modify its API to interact with the Semantic repository enabler to manage the trained models (store and get them), being the latter a dedicated solution for that purpose.	UPV – SRIPAS	-	Completed
18	**FL Training Collector enabler**	The FL Training Collector has been fully integrated with the FL Repository and the FL Local Operations, allowing for download and storage of models and custom components in the case of FL Repository and for configurable FL training and evaluation in the case of FL Local Operations. Proper connections can be established with the FL Orchestrator, but in order to fully integrate the system some internal changes in FL Orchestrator are needed. These integrations will be then necessary for the proper functioning of Pilot 3b.	UPV – SRIPAS	P3b	Completed
19	**FL Repository enabler**	The FL Repository has been fully integrated with the FL Orchestrator, FL Training Collector and the FL Local Operations. The integration allows for flexible ML model, training results and custom modules download and storage. These integrations will be then necessary for the proper functioning of Pilot 3b.	SRIPAS – PRO	P2, P3b	Completed
20	**FL Local Operations enabler**	The FL Local Operations has been fully integrated with the FL Training Collector and the FL Repository, allowing for the configurable start and monitoring of the FL training process and the flexible ML model and custom component loading respectively. The integration with the FL Orchestrator still necessitates some internal updates in the code. These integrations will then be used in pilots 3b and 2.	SRIPAS – PRO	P3b, P2	Completed
21	**MR enabler - EDBE**	The MR enabler receives messages in real time, through topics that it has subscribed, every time that the EDBE publishes a message to the specific topic (mqtt protocols). The message could contain either information for a real time alert, or an update on the location of the construction's workers.	ICCS	P3b, P2	Completed

#	Enablers Involved	Integration Description	Partner Responsible & Involved	Pilots Involved	Current status
22	**MR enabler - Tactile dashboard**	Integration of the interface holding the updatable fields for the MR enabler inside the PUI9 framework. Those fields will be configurable through the PUI9 framework and will be received by the MR enabler through a REST API call, when the MR enabler is executed.	ICCS – PRO	P2	Completed
23	**MR enabler - PUD**	The MR enabler receives data, containing health metrics, from the Hololens device that is executed on, through constant REST API calls, and stores them. Then, the PUD enabler make constant REST API calls to receive the latest values and display them.	ICCS	P2	Completed
24	**MR enabler - LTSE**	Retrieve workers' data from the LTSE's database and display them on MR through REST API calls. Also prepare reports on the MR enabler's UI and send them to the LTSE database through REST API requests.	ICCS	P2	Completed
25	**SDN controller - ACN enabler**	Integration of SDN controller and ACN enablers is for collecting information from SDN controller and use controller to configure (rerouting paths) the network, to execute the results of optimisation from AI module into the SDN network. ACN modules are using ONOS applications: Path Generator and Maintainer in ACN communicates directly with ONOS and independently with two ONOS build-in applications, i.e., Intent Forwarding (IFWD) and Intent Monitoring and Rerouting (IMR).	OPL	P2	Completed
26	**Tactile dashboard - IdM**	Integration of the tactile dashboard and the IdM in order to authenticate users by means of the stored tokens in the IdM.	PRO – S21Sec	ALL	Completed
27	**Tactile dashboard – Authorisation enabler**	Integration of the tactile dashboard and the Authz in order to authorized specific roles to specific users in the dashboard.	PRO – S21Sec	ALL	Completed
28	**Multi-link – LTSE**	The Multi-link enabler makes use of VPN tunnels and dedicated information to perform. Similarly to the VPN-LTSE integration, the multi-link enabler data will be stored in the LTSE, thus requiring some modifications of its API to be implemented.	UPV	ALL	Completed
29	**LTSE – BKPI**	Visualization of historical data from Malta Freeport stored in the NoSQL part of the LTSE through the BKPI enabler	PRO	P1	Completed
30	**PUD – BKPI**	PUD's underlying technology uses Grafana as main representation framework. Project envisions BKPI as the	ICCS	ALL	Completed

#	Enablers Involved	Integration Description	Partner Responsible & Involved	Pilots Involved	Current status
		main one, and therefore some dedicated dashboards and plugins (as data are to be stored in LTSE) are needed			
31	**FL Local operations & DLT-based FL**	A custom FL training strategy will store aggregated weights along with singular client weights in the FL-DLT enabler in order to later retrieve from the FL-DLT computed client reputation scores. More precisely, the FL Local Operations should send a series of requests, first sending the files containing the aggregated weights, then client weights, and then finally a JSON file with metadata like the round index and number of clients the FL-DLT should wait for. Then the FL Local Operations should, on sending an HTTP request with a previously specified format, receive a JSON response with the reputation score for a given client. This integration will not be used in any pilots.	SRIPAS - CERTH	P3b	Pending
32	**Cybersecurity agents and server**	Cybersecurity Server has 2 parts, one the Incident Response must be installed in the cloud and the Incident detection can be implement in the cloud or the edge (Depends on each Pilot). Finally, the agent needs a Linux or Windows compatible OS.	S21Sec	ALL	Completed
33	**BKPI – Tactile Dashboard**	BKPI enabler is essential to visualized time-series KPI pilot data, and therefore it was natural to integrate them with the main user framework of the project. To that end, the development guides of the tactile dashboard enabler were leveraged to generate such interfaces, integrating them in the same solution, avoiding the need of deploying different web applications independently.	PRO	P1	Completed

2.3. End-to-end testing

Subsequent to unit, functional and integration testing, end-to-end testing phase summarises the evaluation of the solution's behavior as a whole and ensures that the entirety of its pipeline behaves as expected. ASSIST-IoT facilitates an array of trials and sub-trials that serve as testbed for the execution of end-to-end testing.

To address this task methodically and ensure its successful implementation, certain assumptions need to be taken into account. The pilot trials are divided among the constituent enablers. Integration testing's focal point was the test of integration of couples of enablers' connections. If each couple of enablers is conceived as a sub-system of the application, it is rational that if any of the sub-system fails, then the whole application fails.

Hence, the significance lies in validation the seamless operation of all application sub-systems. The purpose of conducting end-to-end testing is to make sure that all sub-systems function as expected, commencing from the initial IoT input data and ending with the final output of each trial. This is the best way to safeguard the overall reliability and functionality of the application.

In ASSIST-IoT, the aforementioned sub-systems collectively form each application trial and sub-trial. By subjecting these sub-systems under test, the purpose of minimising the risk of failure is effectively accomplished. The following tables provide detailed reports on the tests conducted for each corresponding trial and sub-trial.

2.3.1. Pilot 1: Port Automation

2.3.1.1. Trial #1: Tracking assets in terminal yard

Table 2: End-to-end testing of Pilot 1 Trial 1

Pilot 1						
Trial #1						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
GWEN - EDBE	Edge devices send telemetry data in JSON format via MQTT client to the EDBE	JSON including location and telemetry data	JSON including location and telemetry data is sent in an MQTT topic	Yes	Pass	Manual
				Phase 1		-
EDBE - LTSE	EDBE MQTT topics are retrieved by the MQTT client acting as subscriber in the cloud server, and forwarding the long-term historical data to the noSQL cluster of the LTSE	JSON including location and telemetry data	JSON including location and telemetry data is stored in LTSE	Yes	Pass	Manual
				Phase 1		-
LTSE - BKPI	The BKPI visualizes in charts/graphs the time-series and historical data stored in the LTSE	Cranes' telemetry and TOS	A user can see a graph composed with relevant pilot data	Yes	Pass	Manual
				Phase 1		-
BKPI and PUD with Tactile dashboard	All the GUIs of the pilot are available in a single interface	Business-oriented logs and devices performance logs	BKPI and PUD GUIs are embedded into the tactile dashboard	Yes	Pass	Manual
				Phase 1		-
Authorisation, IdM, Tactile Dashboard	Tactile dashboard enabler interacts with the Authorisation and Identity manager enablers, deferring to them authentication and authorization of the users	User credentials	User identity and permissions are granted/ declined	Yes	Pass	Manual
				Phase 1		-

2.3.1.2. Trial #2: Automated CHE cooperation

Table 3: End-to-end testing of Pilot 1 Trial 2

Pilot 1						
Trial #2						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Authorisation, IdM, on Edge devices	GWEN and Edge devices of the pilot have the Authorisation and Identity manager enablers installed on it, deferring to them authentication and authorization of the machine-to-machine communication	User credentials	User identity and permissions are granted/declined	NO	Pending	*Manual*
				After trials on the port site		-
Cybersecurity monitoring agent and Cybersecurity monitoring enablers	The cybersecurity monitoring agent reads multiple OS logs from edge devices, which are sent to the cybersecurity monitoring agent, who in turn analyses potential threats over the pilot infrastructure	Log of incidents and other events	Logs are handled by the cybersecurity agent in the cloud server	NO	Pending	*Manual*
				After trials on the port site		-

2.3.1.3. Trial #3: RTH remote control with AR support

Table 4: End-to-end testing of Pilot 1 Trial 3

Pilot 1						
Trial #3						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Multi-link with pilot-specific artifacts	The videos captured by the cameras of the ROS of the cranes are transmitted by a couple of network access managed by multilink enabler	RTMP video files	RTMP video files are received without QoS reduction	NO	Pending	Manual
				After trials on the port site		-

Pilot 1						
Trial #3						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Video Augmentation enabler with pilot-specific artifacts	The videos captured by the cameras of the ROS of the cranes are enhanced with the visual graphics inferenced by the video augmentation enabler	RTMP video files	RTMP video files are overlapped with bounding boxes from Video Augmentation enabler inference	NO	Pending	Manual
				After trials on the port site		-

2.3.1. Pilot 2: Smart safety of workers

2.3.1.1. Trial #1: Occupational safety and health monitoring

2.3.1.1.1. Sub-trial #1: Workers' health and safety assurance sub-trial

Table 5: End-to-end testing of Pilot 2 Trial 1, sub-trial 1

Pilot 2						
Trial #1, Sub-trial #1						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Authorisation, Identity Manager, Tactile Dashboard	Tactile dashboard enabler interacts with the Authorisation and Identity manager enablers, deferring to them authorisation and authentication of the users	User credentials	User identity and permissions	Yes	Pass	Manual
				After trials on the construction site		Also tested in other P2 trials.
Workplace safety controller (custom), Edge Data Broker, Tactile Dashboard	The Workplace safety controller issues notifications, EDBE distributes them to TD, TD displays them in real time	Notification	Graphical user interface	Yes	Pass	Manual
				On the construction site		All notification types were tested in real scenarios. Also tested in other P2 trials.

Pilot 2						
Trial #1, Sub-trial #1						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Edge Data Broker, Semantic Annotation, Semantic Translation, Workplace safety controller (custom)	Sensor is annotated by SemAnn, optionally translated by SemTrans. All data flows through EDBE	Sensor data	Annotated & translated semantic sensor data	Yes	Pass	Manual
				On the construction site		All sensor input types were tested in real scenarios. Also tested in other P2 trials.
Device interfaces (custom), Edge Data Broker	Device interfaces can send sensor data to EDBE and receive actuation instructions	Sensor data, actuation instructions	Sensor data, actuation instructions	Yes	Pass	Manual
				On the construction site		Also tested in other P2 trials.
Tactile Dashboard, Long Term Storage	The manager is able to modify worker and device data in LTSE	User controls	Stored user & device configuration	Yes	Pass	Manual
				After trials on the construction site		Also tested in other P2 trials.
Construction site controller (custom), Workplace safety controller (custom), Semantic Annotation, Long Term Storage, Edge Data Broker	The updated worker and device configuration is propagated via EDBE to SemAnn to be annotated and then to the Construction site controller. Workplace safety controller retrieves this information	Stored user & device configuration	Semantic worksite configuration	Yes	Pass	Manual
				On the construction site		Also tested in other P2 trials.
Semantic Repository, Workplace safety controller (custom)	The Workplace safety controller retrieves the needed ontologies from SemRepo	Stored ontologies	Ontologies	Yes	Pass	Manual
				On the construction site		Also tested in other P2 trials.
Long Term Storage, Business KPI Reporting	BKPI displays information logged in LTSE	Log of incidents and other events	Graphical user interface	Yes	Pass	Manual
				On the construction site		Also tested in other P2 trials.

2.3.1.1.2. Sub-trial #2: Geofencing boundaries enforcement sub-trial

Table 6: End-to-end testing of Pilot 2 Trial 1, sub-trial 2

Pilot 2						
Trial #1, Sub-trial #2						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Edge Data Broker, Location Tracking, Location Processing	Location Tracking enabler sends location data to Location Processing enabler via EDBE	Location data	Query outputs from the Location Processing enabler	Yes	Pass	Manual
				On the construction site		Also tested in other P2 trials.
Long Term Storage, Integrity Verification. Tactile Dashboard, Workplace safety controller (custom)	Workplace safety saves incident logs in LTSE and records them in the Integrity Verification enabler. TD can retrieve the log and verify its integrity	Incident log	Graphical user interface	Yes	Pass	Manual
				On the construction site		Also tested in other P2 trials.
Semantic Repository Enabler, BIM processor (custom), Location Processing	SemRepo notifies the BIM processor about changes to the BIM model. The BIM processor extracts the new danger zones and stores them in LocProc	BIM model	Danger zones stored in LocProc	Yes	Pass	Manual
				On the construction site		–
Workplace safety controller (custom), Location Processing, Edge Data Broker	Workplace safety controller issues queries to LocProc and gets responses. Data flows through EDBE	Spatial query	Spatial query response	Yes	Pass	Manual
				On the construction site		Also tested in Trial #1, Subtrial #3 and Trial #3.

2.3.1.1.1. Sub-trial #3: Construction site access control sub-trial

Table 7: End-to-end testing of Pilot 2 Trial 1, sub-trial 3

Pilot 2						
Trial #1, Subtrial #3						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Image collector (custom), Amazon Rekognition (external), Edge Data	Image collector collects images from the camera,	Camera images	Information about detected people and	Yes	Pass	Manual

Pilot 2						
Trial #1, Subtrial #3						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Broker, Workplace safety controller (custom)	preprocesses them and sends to Amazon Rekognition. The results are then sent via EDBE to the Workplace safety controller		their PPE	On the construction site		–

2.3.1.2. Trial #2: Fall-related incident identification

Table 8: End-to-end testing of Pilot 2 Trial 2

Pilot 2						
Trial #2						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Workplace safety controller (custom), FL Local Operations	Workplace safety controller constructs an inference request and sends it to FL Local Operations via a gRPC stream. FL LocOps sends back a streaming reply with the inference results	FL model feature vector	Inferred output of the model	Yes	Pass	Manual
				On the construction site		-

2.3.1.3. Trial #3: Health and safety inspection support

Table 9: End-to-end testing of Pilot 2 Trial 3

Pilot 2						
Trial #3						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Mixed Reality, Edge Data Broker, Workplace safety Controller	MR enabler receives notifications from the Workplace safety controller via EDBE. The notifications are shown to the user.	Notifications	Mixed reality interface	Yes	Pass	Manual
				On the construction site		-

Pilot 2						
Trial #3						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Mixed Reality, Performance and Usage Diagnosis, Tactile Dashboard	The configuration of the MR device can be controlled in TD. The MR device reports its metrics to PUD	Configuration, metrics	Metrics	Yes	Pass	Manual
				On the construction site		–
Mixed Reality, Semantic Repository	The MR enabler retrieves the needed 3D models from SemRepo and saves captured photos there	Stored 3D models / captured photos	3D models / stored photos	Yes	Pass	Manual
				On the construction site		–
Mixed Reality, Long Term Storage	The MR enabler retrieves worker information from LTSE and stores there the safety reports	Stored worker information / safety reports	Worker information / stored safety reports	Yes	Pass	Manual
				On the construction site		–
Mixed Reality, Location Tracking, Location Processing, Edge Data Broker	The locations of the MR device's user and other workers are processed in real-time by the Location Processing enabler to determine the locations of worker around the manager. The information is passed to the MR device via EDBE	Disaggregated worker location	Disaggregated worker location	Yes	Pass	Manual
				On the construction site		–

2.3.2. Pilot 3A: Vehicle in-service emissions diagnostics

2.3.2.1. Trial #1: Fleet in-service emission verification

Trial #1 from Pilot 3A can be represented with the following figure (Fig. 1), which represents the interactions of the enablers and custom developments implemented for addressing the pilot.

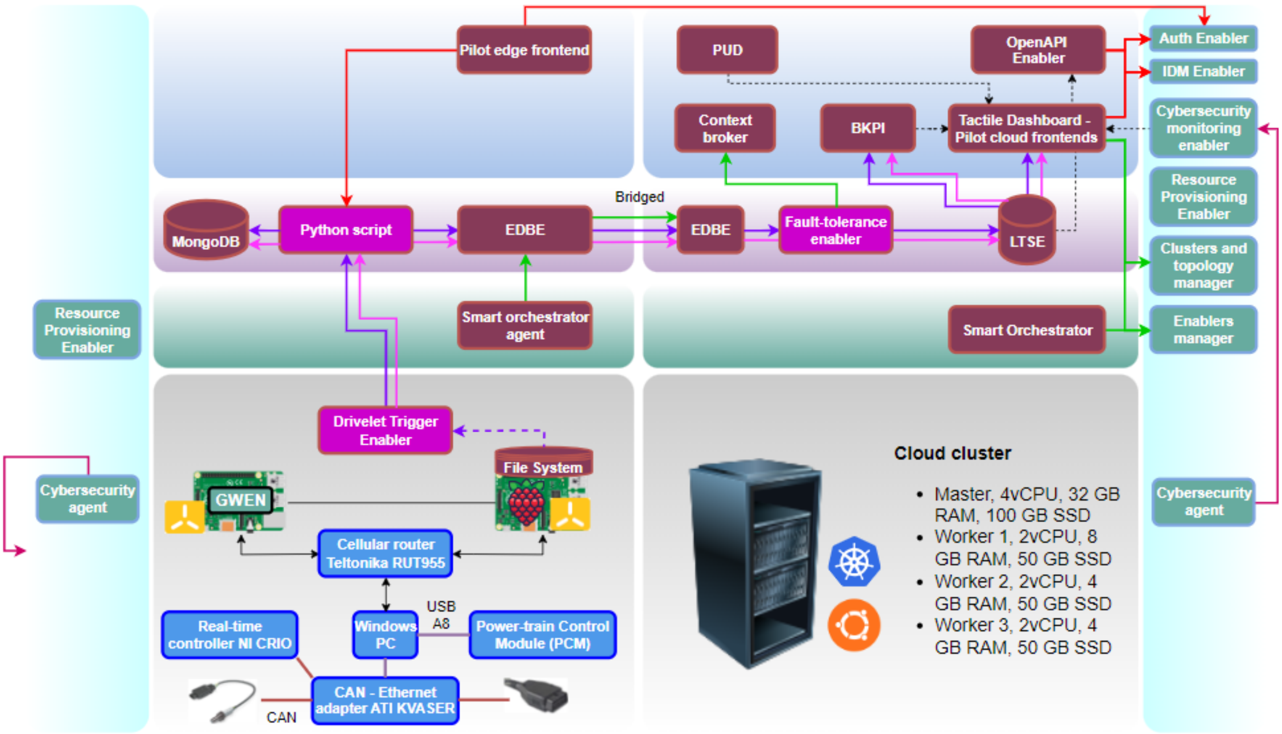


Figure 1. Pilot3A Trial #1 Architectural diagram

Table 10: End-to-end testing of Pilot 3A Trial 1

Pilot 3A						
Trial #1						
Enablers connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Tactile dashboard with BKPI, PUD Cybersecurity monitoring and Manageability enablers	Several frontends can be consumed by users. They have to be accessed from a single point, the tactile dashboard	N/A	Logged user can access to their respective frontends	Yes	Pass	Manual
				Phase 2		-
BKPI with LTSE	The BKPI requires connectivity to the LTSE to work	Pilot data (e.g., sensor data)	A user can see a graph composed with relevant pilot data*	Yes	Pass	Manual
				Phase 2		-
OpenAPI with LTSE	LTSE's API published in the OpenAPI. Connecting between them should be working	API doc	A user can interact with the LTSE using the OpenAPI*	No	-	Manual
				-		-
EDBE with context broker	The context broker is populated with MQTT data coming from the EDBE	Data from the smart orchestrator	A user can check the groups available and	Yes	Pass	Manual

Pilot 3A						
Trial #1						
Enablers connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
		agent	the status of the enablers deployed	Phase 3		-
Smart orchestrator with clusters and topology manager, and enablers manager	The smart orchestrator and the manageability enablers must have connection and compatible versions to support clusters and enablers management	API calls with related data	A user register a car in a group and deploy enablers in such group	Yes	Pass	Manual
				Phase 2		-
EDBE with LTSE	EDBE and LTSE must have connection so that pilot data can be persisted	Pilot data	User can request pilot data using the LTSE's API*	Yes	Pass	Manual
				Phase 2		-
Auth enabler with pilot edge frontend and EDBE	Car user can accept or deny that its geolocation data are persisted	Geolocation data	User can request geolocation data using the LTSE's API	Yes	Pass	Manual
				Phase 3		-
Cybersecurity monitoring enabler with cloud and edge agents	Connectivity between server and agents must be in place so that events can be analysed	Cybersecurity logs	Logs can be checked from the cybersec. frontend	No	Pass	Manual
				-		-
Smart orchestrator agent with EDBE	To manage the car's edge node, there must be an active connection between the agent and the cloud's EDBE, bridged to the car's	Data about the lifecycle status of the enablers deployed in the car	A user can check the car registered and the status of the enablers deployed in the context broker	Yes	Pass	Manual
				Phase 3		-
Drivelet trigger enabler and Python script	This pilot enablers must be connected so that new data are processed by the script	Drivelets, sensor data	A user can check pilot data from the BKPI frontend*	Yes	Pass	Manual
				Phase 2		-
Python script and MongoDB	The script needs to process data and store them locally	Pilot data	A user can check pilot data from the BKPI frontend*	Yes		Manual
				Phase 2		-

2.3.3. Pilot 3B: Vehicle exterior condition inspection and documentation

2.3.3.1. Trial #1: Vehicle exterior condition inspection and documentation

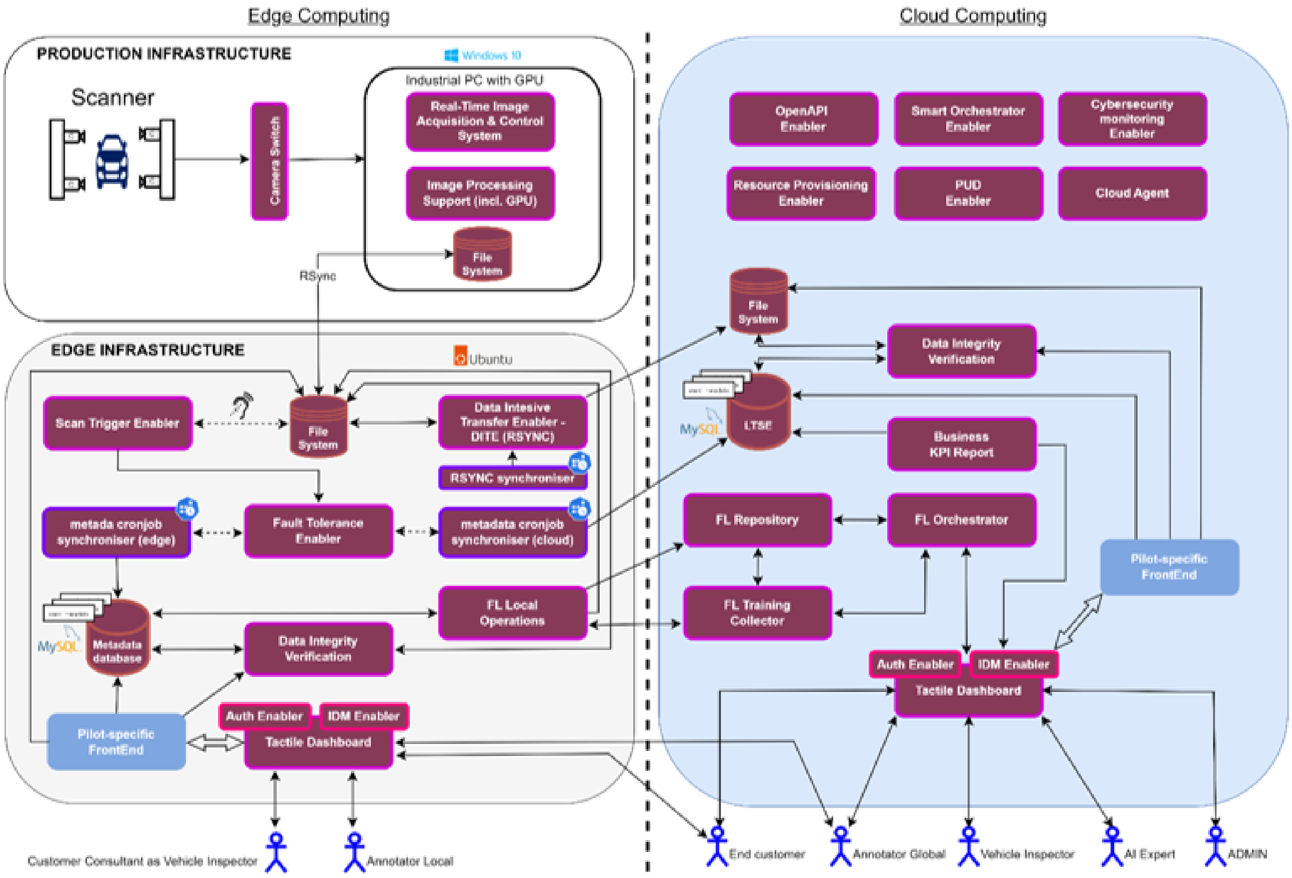


Figure 2. Pilot 3B Trial #1 Architectural diagram

Table 11: End-to-end testing of Pilot 3B Trial 1

Pilot 3B						
Trial #1						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Production infrastructure File System with edge infrastructure File System	Files generated by the scanner are synchronously replicated to edge development environment via RSYNC	Folder generated after scanning vehicle	Folder replicated in edge environment.	Yes Phase 1	Pass	Auto
Scan Trigger Enabler with local File System	Enabler to synchronous listening of new folders added in a host path	Folder added in file system	Logs indicating name of folder received	Yes Phase 1	Pass	Auto

Pilot 3B						
Trial #1						
Enabler's connection under test	**Description**	**Input**	**Test condition**	**Already tested / Which phase**	**Test result**	**Automated or manual / Comments in case of manual**
Data Intensive Transfer Enabler between the local file system and the cloud file system	Asynchronous replication of data from edge to cloud environment	Existing files and folders on a specific host path	Folder replicated in edge environment.	Yes Phase 1	Pass	Auto (via cronjob)
Scan Trigger Enabler with Fault Tolerance Enabler	Scan Trigger Enabler detects a new folder with processable data send to Fault Tolerance Enabler	Folder added in the filesystem with processable data	New files and folders on a specific host path	Yes Phase 1	Pass	Auto
Fault Tolerance Enabler with edge metadata database	When new data in JSON format is received on a specific topic, the data is parsed and sent to the metadata database	JSON data in a specific topic	New files and folders on a specific host path	Yes Phase 1		Auto (via cronjob)
Fault Tolerance Enabler with cloud LTSE	When new data in JSON format is received on a specific topic, the data is parsed and sent to the cloud LTSE	JSON data in a specific topic	Metadata processed in JSON format	Yes *Phase 1*	Pass	Auto (via cronjob)

2.4. Acceptance testing

After end-to-end testing phase, which ensures the functional robustness of the final solution, the next crucial step is acceptance testing. At this stage, the project has reached its development conclusion, and it is imperative to evaluate the delivered final solution.

This aligns with the previously introduced acceptance testing strategy outlined in the prior version of the deliverable D6.3 [1.]. As detailed in that version, acceptance testing's backbone is the requirements initially outlined in D3.3 [5.]. This involves aligning the initial specifications envisioned by the project partners with the final solutions developed to fulfil those specifications.

To effectively verify and validate these requirements, a careful classification process was carried out for both pilot and common requirements. This classification considered factors such as the specific pilot to which each requirement pertained, its priority, and whether it was a functional or non-functional requirement. The resulting tables include not only the definitions of these requirements but also their acceptance status and the procedure employed for their validation. For requirements that fell lower in priority or were not met, there is a detailed justification provided for each one. As stated in section 1.5, due to variations in the maturity of the pilots, **pilot-related requirements will be consolidated in D8.3.**

2.4.1. Common requirements

Table 12: R-C-1 Data sovereignty

R-C-1: Data sovereignty (Non – Functional)			
Priority	Must	**Partners responsible**	TL
Rationale	Different data sets are obtained from the pilots		
Description	ASSIST-IoT shall guarantee that in order to take better operational decisions, data quality must be ensured		
Acceptance criteria	Data integration of at least 3 different sources of information must be interoperable.		
Acceptance Status	Pass		
Validation Procedure	A list of different data sources from ASSIT-IoT pilots confirms that different type of information has been collected and interoperated: • Pilot 1: RTG cranes telemetry data, STS cranes telemetry data, Trucks GPS data, Terminal Operating System work instructions data, Video streams • Pilot 2: GPS location data, UWB positioning and geofencing data, BIM model-related data, human skin temperature and heart rate data, weather data, UV radiation data, acceleration data, alerting data, work reports. • Pilot 3A: NOx and CO2 emissions data, PCM data, engine data, edge devices performance logs, network metrics Pilot 3B: Scanned images, scanner metadata, AI serialised models, vehicle exterior defects data		
Additional report	N/A		

Table 13: R-C-2 Data governance

R-C-2: Data governance (Non – Functional)			
Priority	Must	**Partners responsible**	TL
Rationale	Different data sets are obtained from the pilots		
Description	ASSIST-IoT shall guarantee that in order to take better operational decisions, data governance must be ensured		
Acceptance criteria	Data of at least 3 different sources of information has to be managed.		
Acceptance Status	Pass		
Validation Procedure	Data governance aims at ensuring that the entire data lifecycle is secure, private, accurate, available and usable. ASSIST-IoT considers the Edge Data Broker (EDB) and the Long-Term Storage Enabler (LTSE) as main mechanisms to securely transmit, store and offer data. Having them developed and integrated, and having implement them for managing the flow of several sources of information (see R-C-1 above), the requirement is considered validated. In case that ownership and protection of the data is to be ensured, complementary enablers from the trust vertical (i.e., DLT-based enablers) can be added in the flow to that aim. Finally, regarding data quality, pilots have implemented and deployed dedicated services at their respective edge sites to capture and pre-process the incoming data using their particular business logics (e.g., in Pilot 3A, curated daily data are processed before being ingested in the system, and in Pilot 3B, metadata from active scans are pre-processed before being managed by the rest of the data management enablers).		
Additional report	N/A		

Table 14: R-C-3 Compliance with legal requirements on data protection

R-C-3: Compliance with legal requirements on data protection (Non – Functional)			
Priority	Must	**Partners responsible**	TL, FORD-WEKE
Rationale	Protection of personal data in compliance to the GDPR and any other identified legal requirement that is relevant to project scope.		
Description	In any task involving process personal data the following will be required: • copies of ethical approvals for the collection of personal data by the competent National Data Protection Authority, • details on the procedures to be implemented for data collection, storage, protection, retention and destruction and confirmation that the project will comply with the relevant national and EU legislation, • details on the informed consent procedures to be implemented, • confirmation that the existing data is publicly available, or otherwise relevant authorisation.		
Acceptance criteria	Tracked user should be aware of the data that is tracked, how it is being used, for what objectives, and has agreed to it e.g. GPS location data are kept in the far edge node and only regional position will be propagated to the remote system.		
Acceptance Status	Pass		
Validation Procedure	Extensive information about this requirement can be seen in D2.3 (Ethics and privacy protection manual) and D2.4 (updated version of the former), about the project ethics framework, legal aspects, data privacy policies as well as management and mitigation strategies. For pilot participants, which are the people involved in the pilot demonstrations and other event participants, that includes the participants in workshops, webinars, conferences etc. (partner employees fall under the same categories, based on their participation in events).		
Additional report	N/A		

Table 15: R-C-5 Local Processing Capabilities

R-C-5: Local Processing Capabilities (Non – Functional)			
Priority	Must	**Partners responsible**	MOW
Rationale	Transmission of sensitive information shall be kept to a minimum.		
Description	ASSIST-IoT shall be able to process sensitive information at the edge and decide which information needs to be transmitted to a central location for storage or further processing.		
Acceptance criteria	ASSIST-IoT software is designed to support edge processing operations and provides useful features at the edge.		
Acceptance Status	Pass		
Validation Procedure	ASSIST-IoT has several enablers that help with processing the data on the edge. The Edge Data Broker supports data routing on the edge and can be configured to direct the data to be processed locally. The Semantic Annotation enabler supports edge deployments on ARM CPUs and can effectively process semantic data there. The suite of FL enablers (especially the FL Local Operations enabler) allow for machine learning model training on the edge, without transmitting the data to a central location. The manageability enablers have the functionalities for managing multi-cluster deployments, including heterogeneous systems with edge devices. Specific components can be deployed only on the edge devices, ensuring the processing takes place there.		
Additional report	N/A		

Table 16: R-C-6 Data Persistence and Trust

R-C-6: Data Persistence and Trust (Non – Functional)			
Priority	Must	**Partners responsible**	FORD-WEKE
Rationale	The specific kind of data (specific for every pilot) should be retained so that one can demonstrate the integrity and non-repudiation for third party		
Description	ASSIST-IoT shall retain information about: 1. List of all manipulation operations with container 2. All the information referring to worker health and safety 3. The vehicle's emissions profile throughput its lifetime so that it can demonstrate its in-service conformity to a third party. The information of the calibration used for every vehicle and the modifications performed on it will be also kept, as well information of parts being replaced (including sensors) 4. The current rented vehicle conditions at the moment of taking the car should be securely saved (nobody could manipulate any data and pictures) till the time he brings the car back. The difference of the rented vehicle conditions is the basis of his potential duties to the car rental company and the potentially existing or closed insurance contracts		
Acceptance criteria	ASSIST-IoT will handle local data persistence according to system resources. ASSIST-IoT will increase local data persistence when the system is set to Active Monitoring state.		
Acceptance Status	Pass		
Validation Procedure	To fulfil this objective, the project has designed and developed a set of DLT-based enablers, each of them for different purposes, among others, to store in a ledger critical logs and audit results, and to store in a ledger critical. The performance of those enablers have been tested, and they have effectively implemented in pilot 2 and ongoing in pilot 3a and 3b. The information written in a ledger is shared among the peers and cannot be deleted. Hence, not all data or logs are stored, only that finally marked as critical.		
Additional report	N/A		

Table 17: R-C-7 Edge-oriented deployment

R-C-7: Edge-oriented deployment (Non – Functional)			
Priority	Must	**Partners responsible**	TL
Rationale	The deployment of Next Generation IoT systems should not be extremely expensive, and should be scalable		
Description	The ASSIST-IoT architecture should support a scalable solution, which would reduce the cost of initial deployments as much as possible		
Acceptance criteria	Large-scale deployments with hundreds of gateways with a wide variety of different execution environments should be supported. 10% reduction in time for provisioning new elements in the deployment.		
Acceptance Status	Pass		
Validation Procedure	The ASSIST-IoT architecture is scalable by design, and both the technological selections and the specific features brought by several enablers contribute to that aim. This requirement is considered validated after the specific performance tests performed in Pilot 3A, in which a fleet of ~100.000 cars (edge nodes) has been simulated to ensure that enablers can manage the number of connections and traffic load generated. Among specific enablers that contribute to scalability, apart from the inherently scalable characteristics of K8s, one can find the Smart orchestrator, which allows deploying enablers in the register edge nodes (or groups of edge nodes) easily; the EDB, which allows transmitting data from tens of thousands of nodes – considering bridging strategies; and the resource provisioning enabler, which predicts typical peaks of usage to allocate resources to enablers in case they are expected to be needed. With respect to the reduction of time in provisioning new elements, the fact of using a Cloud Native		

R-C-7: Edge-oriented deployment (Non – Functional)	
	approach – considering the pillars of microservices, containers and DevOps, jointly with the dedicated user-friendly dashboards to manage an ASSIST-IoT-based system, ensures fast deployment times in comparison to systems based on SOA and/or VM-based deployments. Still, the specific number has not been captured as it greatly depends on the baseline used.
Additional report	N/A

Table 18: R-C-8 Edge node modularity

R-C-8: Edge node modularity (Non – Functional)			
Priority	Must	**Partners responsible**	ALL
Rationale	A modular design of the Edge Node (or gateway) has many advantages, such as reduction of cost, as the baseline hardware is lower, and flexibility, as complementary hardware can be added later on when needed for supporting specific use cases.		
Description	The ASSIST-IoT Edge Node must support use cases from different verticals, and therefore it should be prepared to incorporate dedicated hardware ad-hoc. Hardware demanded by all pilots (e.g., WiFi module, Ethernet interfaces, SD card slots) will be added in the main board, whereas a set of Board-to-board (B2B) modules will be incorporated to extend the baseline capabilities.		
Acceptance criteria	The Edge Node will be used in the pilots, effectively extending its basic features for those cases that require it (e.g., adding a GPU, extra RAM or a CAN module).		
Acceptance Status	Pass		
Validation Procedure	The Consortium has produced its own gateway, with edge capabilities, branded as *GWEN* (Gateway and Edge Node). The GWEN is composed of standard interfaces used in Industrial sectors, both wired (RS232, RS485, CAN 2.0, CAN FD, Ethernet) and wireless (UWB, WiFi, BLE, WiFi 6 and cellular – 4G-5G). Apart from is baseline processing and storage capabilities, the GWEN has a pair of micro SD slot and a set of expansion modules to increase the hardware capabilities according to the real needs of the target scenario. Besides, a Yocto-based OS is implemented, to effectively manage the hardware and also containing the virtualization technologies needed to run the enablers of the project. Four GWENs have been manufactured and shipped to the pilots, which have given positive feedback of its capabilities.		
Additional report	N/A		

Table 19: R-C-9 Workload placement

R-C-9: Workload placement (Non – Functional)			
Priority	Must	**Partners responsible**	UPV
Rationale	Personnel in charge of deploying the software artifacts in the clusters might not have the necessary knowledge to select their optimal place, especially when there are several nodes and clusters involved.		
Description	The Smart Orchestrator of the ASSIST-IoT architecture will include intelligent mechanisms to place the (software) workloads in the managed equipment. This element will include a set of policies that a user can specify when deploying a workload (manual placement will be also allowed). Some of these policies will be: • Placing workloads in edge nodes (i.e., differentiate edge from cloud resources), • Placing workloads in nodes with larger more free resources, • Placing workloads in nodes with smaller number of free resources.		
Acceptance criteria	At least three policies will be included and validated for automatically selecting the optimal cluster/s to deploy workloads. At least one of them will be supported by AI to make the decision.		
Acceptance Status	Pass		

R-C-9: Workload placement (Non – Functional)	
Validation Procedure	A scheduler component has been developed as a microservice within the Smart Orchestrator enabler. This component is responsible for choosing the optimal cluster for deploying enablers, taking into consideration various policies: • Best-fit: Workloads are assigned to nodes with the least available resources. • Worst-fit: Workloads are assigned to nodes with the most available resources. • Most-traffic: Workloads are placed on nodes with the highest network traffic. To enhance the accuracy of the scheduling process, this component leverages historical data. It selects the best cluster based on predictions derived from the data of the past 7 days, forecasting the next 7 days. All the data is retrieved with the PUD Enabler, the resource monitoring enabler in the ASSIST-IoT architecture. For the correct validation, the different clusters were exposed to varying levels of traffic and computational loads. Enablers were instantiated using the scheduler, ensuring that they were placed in the appropriate cluster.
Additional report	N/A

Table 20: R-C-10 Transmission security

R-C-10: Transmission security (Non – Functional)			
Priority	Must	Partners responsible	UPV
Rationale	Business scenarios likely involve critical and/or personal data that must be protected so they cannot be accessed by unauthorised or malignant users.		
Description	The ASSIST-IoT architecture must support the introduction of tunnelling technologies for encrypting the transmission of information within the network. To that end, VPN technology will be implemented for allowing the connection of external devices (i.e., (not belonging to a site's network infrastructure) to ASSIST-IoT, and SD-WAN technology will be leveraged for connecting clusters from different sites.		
Acceptance criteria	(A set of) external devices will be effectively incorporated to the network site via VPN, and at least two clusters managed by the project will be connected via SD-WAN. Network sniffers will be used to ensure that involved traffic is encrypted.		
Acceptance Status	Pass		
Validation Procedure	ASSIST-IoT provides both a VPN and a SD-WAN solution in the form of enablers. Validation of the first have been done in Pilot 1, whereas SD-WAN suite testing has been performed in lab conditions. To perform the tests, Wireshark was used to confirm that connections were encrypted. Additionally, two additional mechanisms are in place: MQTTS, when using the EDB enabler with certificates, in the cases that additional security is needed; and uTLS between pods transmissions in K8s networks, thanks to the use of a service mesh (Cilium) in the virtualized environments. The encryption capabilities of these two mechanisms have been also tested satisfactorily via sniffers.		
Additional report	N/A		

Table 21: R-C-11 Network optimisation

R-C-11: Network optimisation (Non – Functional)			
Priority	Must	Partners responsible	UPV
Rationale	Usually, network-related requirements tackle bandwidth, latency, and security features. Other aspect that could be taken into consideration is the traffic or application type, to be prioritised.		
Description	The architecture must support dedicated mechanisms to prioritise specific traffic in the network, so bandwidth is optimised and latency reduced (for these applications).		

R-C-11: Network optimisation (Non – Functional)	
Acceptance criteria	Different mechanisms will be targeted, and the verified application of (at least) one of them will mark this requirement as fulfilled: • Policy-based SDN, • SD-WAN traffic shaping, • Dedicated (multi-)cluster rules, based on CNI.
Acceptance Status	Pass
Validation Procedure	There are many enablers that provide specific features related to this requirement. They have been tested and the results are the following: • Policy-based SDN, targeting ensuring QoS of specific traffic type (to ensure latency, bandwidth), has been implemented via the autoconfigurable network enabler. It has been tested in laboratory conditions, using offline video traffic as target. • Security features in real time are provided via (i) a dedicated enabler, the cybersecurity monitoring agent (which monitors the logs produced and aims at detect cybersecurity menaces), and (ii) the use of policies in the CNI-service mesh considered in the K8s implementation, Cilium, which acts also as firewall. Both have been tested in lab and pilot conditions with great results. • Although slightly transversal to the requirement, the Smart orchestrator (which is in charge of managing Cilium rules) has a scheduling policy to deploy workloads in the cluster with most traffic. This can be of interest for some kind of specific applications. • SD-WAN traffic shaping features have not been tested. The acceptance criteria was to consider the requirement passed if at least one of the above mentioned options was passed, therefore, the requirement has been fulfilled.
Additional report	N/A

Table 22: R-C-12 Streaming Latency

R-C-12: Streaming Latency (Non – Functional)			
Priority	Must	**Partners responsible**	SRIPAS
Rationale	The streaming latency reflects as a critical factor when high volume of small messages is transmitted in the Next Generation IoT systems. This property enables or prohibits real-time processing and has high impact on responsivity of systems.		
Description	The ASSIST-IoT architecture should provide an efficient solution, to stream high volume of messages with low latency and high throughput, enabling high quality data transmission.		
Acceptance criteria	Steaming annotation and translation latency between reception of a message at input topic, and processed message arriving at the output topic should be below <10ms		
Acceptance Status	Pass		
Validation Procedure	The message latency is defined as the amount of time it takes to process a single message (averaged over a period of time) at the core message processing component, without taking into account broker throughput. This is done in order to isolate the software from as many external factors as possible. Even though the message processing enablers are designed to support high volume of small messages, the latency will be tested with messages of small and medium size, to stretch the limits of the simulation. Measurements will be taken over 1 minute of constant message output from the broker, with the target of average processing latency less than 10ms per message.		
Additional report	N/A		

Table 23: R-C-13 Streaming Client Numbers

R-C-13: Streaming Client Numbers (Non – Functional)			
Priority	Must	**Partners responsible**	SRIPAS
Rationale	Next Generation IoT systems are highly distributed and must often support a high number of users at the same time.		
Description	ASSIST-IoT should provide a scalable message stream, that are able to handle many individual clients at the same time, and process messages in parallel, where (provided hardware resource availability) high parallelism does not significantly impact processing latency		
Acceptance criteria	Streaming semantic annotation and translation should both support at least 10 parallel translation or annotation channels (channels may have many clients receiving and sending messages).		
Acceptance Status	Pass		
Validation Procedure	Performance of supported client parallelism is measured by creating a number of streaming channels, and scaling the number up, until resources on the test machine are saturated. Because the number of clients per channel is scaled on the level of the broker, each channel will have just one sender and receiver. Adding more clients per channel is supported (and even expected in a typical use-case) but would make the simulation more dependent on performance of the broker, and not the core of the software under validation. A conservative target is placed at 10 parallel channels, as that is the expected maximum number of channels in a single real-world deployment that will be realistically required.		
Additional report	N/A		

Table 24: R-C-14 File Size Support

R-C-14: File Size Support (Non – Functional)			
Priority	Should	**Partners responsible**	SRIPAS
Rationale	Growth in usage of data in IoT resulted not only in higher volume of network traffic, but also in an increase in the range of file sizes that are considered in IoT. Although IoT messages are very often small, some use cases (e.g. storing and exchanging BIM models) require support for large files.		
Description	The ASSIST-IoT is intended to manipulate and store large files that define data models or that facilitate data transformations, such as ontologies, schema files, and semantic alignment files.		
Acceptance criteria	The ASSIST-IoT semantic repository should support large file sizes, coming from real-world applications or pilots		
Acceptance Status	Pass		
Validation Procedure	During Pilot 2 validation activities, several large (>50MB) BIM files were successfully uploaded to the Semantic Repository enabler, representing a real use case. During laboratory testing, test files of size of over 5GB were successfully uploaded and downloaded. No performance or stability issues were observed.		
Additional report	N/A		

Table 25: R-C-15 External access to resources

R-C-15: External access to resources (Non – Functional)			
Priority	Must	**Partners responsible**	UPV
Rationale	The developments and advancements coming from ASSIST-IoT shall assist open-callers to build new projects under NGIoT architectures.		
Description	The ASSIST-IoT architecture must have an API manager that will act as a gateway/entry point for all the internal services that can be accessed by 3rd party authorised actors. Software artifacts should have their APIs published in the manager so they can be later on consumed by authorised users.		

R-C-15: External access to resources (Non – Functional)	
Acceptance criteria	An external user can consume the API of an internal software resource (if authenticated in the platform and authorised to access to it).
Acceptance Status	Pass
Validation Procedure	The functionality was made available through the OpenAPI enabler, which serves as an API Gateway, enabling third-party users to access resources that were previously designated as accessible from outside the ASSIST-IoT architecture. Access to these resources is provided to users who have been granted permissions in the Identity Manager Enabler. This Identity Manager Enabler is integrated with the OpenAPI Enabler to facilitate access for authorised users to specific resource paths. For making the enabler easy to use, a frontend and backend was created. To make the enabler user-friendly, both a dedicated frontend and backend were developed.
Additional report	N/A

Table 26: R-C-16 Resource monitoring

R-C-16: Resource monitoring (Non – Functional)			
Priority	Must	**Partners responsible**	PRO
Rationale	A NGIoT environment would most likely consist of (almost every time) of several hardware computing nodes, each of them with different workloads deployed. Having a dedicated monitoring stack supported by graphical tools is essential so administrators have an overall picture of the situation, and hence for detecting misuse and/or lack of resources.		
Description	The ASSIST-IoT architecture must provide dedicated mechanisms for gathering resource data from the infrastructure (clusters, mainly RAM and CPU) and from the software artifacts deployed in top of it, Once collected, data will be stored in a dedicated space of the data persistence mechanism, and users will have the option to access it either via commands or dedicated, intuitive graphical dashboards.		
Acceptance criteria	A user will be able to access to the collected data via dedicated Graphical User Interfaces.		
Acceptance Status	Pass		
Validation Procedure	ASSIST-IoT consists of two components for resource monitoring within each of the clusters: the Performance and Usage Diagnosis Enabler and the PUD Federate. The Performance and Usage Diagnosis (PUD) Enabler provides users with access to RAM, CPU, and traffic data, representing this information through a graphical tool. Furthermore, this data can be ingested by the PUD Federate, a centralized resource monitoring enabler designed to collect data from various PUD Enablers located in different clusters. This integration is automated and tested through the Smart Orchestrator Enabler, which dynamically adds the addresses of these enablers to enable access by the PUD Federate.		
Additional report	N/A		

Table 27: R-C-17 Time-series data visualisation

R-C-17: Time-series data visualisation (Non – Functional)			
Priority	Must	**Partners responsible**	PRO
Rationale	A NGIoT deployment will collect huge amount of data from the unevenly distributed and heterogeneous sources. Among all types of data, time series data (e.g., data from sensors) is becoming the most widespread. Unfortunately, not only collecting, storing, but also analysing massive amounts of this data demands of intuitive visualization tools for providing sufficient analytics needs.		

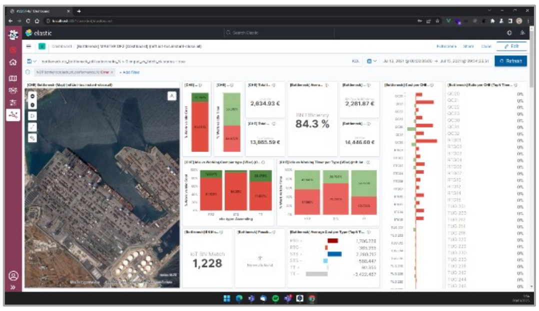
R-C-17: Time-series data visualisation (Non – Functional)	
Description	The ASSIST-IoT architecture must provision dedicated mechanisms that allow for rapidly creating dashboards that pull together charts, maps, histograms, and filters to display the full picture of stakeholder's data. It should also be flexible enough to allow building customised dashboard that enable deeper analysis.
Acceptance criteria	A user will be able to access to the collected time-series data via dedicated Graphical User Interfaces.
Acceptance Status	Pass
Validation Procedure	Project partners have included into ASSIST-IoT enablers' list, the so-called Business KPI reporting enabler. This enabler provides the visualisation needs associated to R-C-17, by easily integrated with the NoSQL cluster of the Long-Term Storage enabler. A couple of graphical examples are provided below. 
Additional report	In case that you ignored this requirement for any reason, justify it.

Table 28: R-C-18 Support for autonomous processing

Requirement: R-C-18, Support for autonomous processing (Non – Functional)			
Priority	Must	**Partners responsible**	SRIPAS
Rationale	In large-scale highly distributed IoT-based systems human involvement in management should be minimised.		
Description	In ASSIST-IoT powered systems decisions that can be taken automatically should be autonomous thanks to self-* mechanisms.		
Acceptance criteria	During the deployment of the pilots, the developments will make use of some of the self-* enablers to execute autonomous operations/decisions		
Acceptance Status	Pass		
Validation Procedure	The following self-* mechanisms have been already successfully applied in some of the pilots: resource provisioning, self-healing and location processing enabler, allowing the respective systems to have location context, autonomy for allocating resources in case a peak or fall of demand is expected and recovery against complete failures. The monitoring and notifying enabler and the automated configuration enabler will be validated in the next months. Additionally, other enablers such as the Smart orchestrator (in which user can let decide the optimal place of the computing continuum to deploy enablers) have also implemented autonomous decision-making processes.		
Additional report	N/A		

Table 29: R-C-19 Support for self-aware systems

R-C-19: Support for self-aware systems (Non – Functional)			
Priority	Must	**Partners responsible**	SRIPAS

R-C-19: Support for self-aware systems (Non – Functional)	
Rationale	In a large-scale highly distributed IoT-based systems, environment awareness and diagnosability should be supported.
Description	ASSIST-IoT powered systems should be able to understand its state, problems and required actions.
Acceptance criteria	Self-awareness capabilities should be utilised in the pilot developments.
Acceptance Status	Pass
Validation Procedure	Three self-* enablers, self-healing, resource provisioning, and monitoring and notification, can manage status-related information of the system, with the purpose to either initiate remediation actions (in the case of the two former) or notify these issues to administrators or to other enablers. At this moment, the self-healing and the resource provisioning enabler have been tested in the pilots and therefore the requirement is considered as fulfilled, whereas the monitoring and notifying enabler is scheduled to be deployed on them soon.
Additional report	N/A

Table 30: R-C-20 Support for system self-healing

R-C-20: Support for system self-healing (Non – Functional)			
Priority	Must	**Partners responsible**	SRIPAS
Rationale	In a large-scale highly distributed IoT-based systems, anomalies should be treated locally and autonomously.		
Description	ASSIST-IoT powered systems should be able to discover and fix abnormal behaviours.		
Acceptance criteria	During the execution of the pilots, self-healing capabilities should characterise the majority of the deployments.		
Acceptance Status	Pass		
Validation Procedure	Due to the use of K8s as key part of the ASSIST-IoT architecture, self-healing capabilities are inherently present in the pilots. Additionally, the self-healing enabler developed in the project allows automatic response to IoT device issues, thus providing self-awareness.		
Additional report	N/A		

Table 31: R-C-21 Reduction of computing demands for AI training

R-C-21: Reduction of computing demands for AI training (Non – Functional)			
Priority	Must	**Partners responsible**	PRO
Rationale	The recent deployments and advancements in artificial intelligence consume large amount of processing resources.		
Description	Within the ASSIST-IoT, distributed AI techniques will be utilised to reduce the communication load and needed processing power.		
Acceptance criteria	At least, 2 pilots of the project will perform ASSIST-IoT AI functionalities on their edge-devices, rather than relying on public cloud.		
Acceptance Status	Pass		
Validation Procedure	The distributed AI functionalities of ASSIST-IoT are provided by the FL system enablers developed under T5.2. As it can be read in Section 2.3.2, on the one hand, pilot 2 Trial #2: Fall-related incident identification have made use of the FL Local Operations for detecting workers' fall accidents, while, on the other hand, pilot 3B Trial #1: Vehicle exterior condition inspection and documentation have made use of the whole FL		

R-C-21: Reduction of computing demands for AI training (Non – Functional)	
	system enablers for the edge-oriented AI-based support for automatic inspection of vehicle surface damages.
Additional report	In case that you ignored this requirement for any reason, justify it.

Table 32: R-C-22 Support for data privacy during the training process

R-C-22: Support for data privacy during the training process (Non – Functional)			
Priority	Must	**Partners responsible**	SRIPAS
Rationale	In many real-life scenarios there is a need to preserve sensitive data at local parties (without sharing) to reduce the risk of leakage.		
Description	In ASSIST-IoT FL architecture should be promoted to conduct the training process in a distributed and secure way without the need to move the data.		
Acceptance criteria	During the continuous training and execution of FL-related use cases, secure and distributed training processes should be utilised.		
Acceptance Status	Pass		
Validation Procedure	In order to fulfil this requirement, two methods adding privacy support to the aggregation mechanism in Federated Learning have been added to the system: homomorphic encryption and differential privacy. Aside from the data privacy inherent in a proper implementation of a FL system (local data never leaves the client), the system can be prompted to use homomorphic encryption (a technique that allows for the aggregation of fully encrypted weights, but due to its current computational complexity can only be used for very small models) or differential privacy (noise is added to the local weights in order to provide additional privacy). The validation procedure then consists of configuring the training with the proper parameters (the training parameters of the FL Training Collector as well as the FL Local Operations need to include either the keyword "homomorphic" for homomorphic encryption or "dp-adaptive" for differential privacy along with the dictionary with the configuration of the privacy mechanism under "privacy-mechanisms"), prompting for homomorphic encryption or differential privacy to be used. Then, the use of privacy mechanism can be observed in the logs of the enablers and final training results metadata.		
Additional report	N/A		

Table 33: R-C-23 Multi-model FL support

R-C-23: Multi-model FL support (Non – Functional)			
Priority	Must	**Partners responsible**	SRIPAS
Rationale	Federated learning should support various techniques, algorithms and models that can be applied on distributed nodes.		
Description	ASSIST-IoT shall be able to support multiple ML algorithms and models suitable for problems and data. ASSIST-IoT should promoted model-agnostic FL architecture.		
Acceptance criteria	At least 2 ML algorithms will be used for the FL training.		
Acceptance Status	Pass		
Validation Procedure	The validity of this statement can be confirmed through the use of FL Repository and subsequent training configuration. An ML model (Convolutional Neural Network for multiclass classification) can be loaded out-of-the box from the FL Repository and trained using the appropriate configuration. FL support is also offered for the model used in Pilot 3b (a Region-based Convolutional Neural Network), which can be later verified from the logs used in that Pilot. Finally, the FL system supports the addition of an arbitrary neural network through saving it in the FL Repository models collection, after which it can be freely used for		

R-C-23: Multi-model FL support (Non – Functional)	
	training in the FL system.
Additional report	N/A

Table 34: R-C-24 Cooperative ML training support

R-C-24: Cooperative ML training support (Non – Functional)			
Priority	Should	**Partners responsible**	PRO
Rationale	The late advancements in AI/ML have led to an increasing interest from different industries to provide these functionalities over their systems. However, in order to support very accurate ML models, huge amount of data should be collected for training. Given the lack of resources, the concept of coopetition (the act of cooperation between competing companies) has gaining momentum, which by means of FL systems, allow competing companies to obtain a common trained model without sharing their data.		
Description	ASSIST-IoT will implement a custom private FL system able to support a coopetition environment, allowing different parties to collaborate in an ML training process without sharing their data.		
Acceptance criteria	At least 10 simultaneous local parties/clients will be used to train an AI model with ASSIST-IoT FL system		
Acceptance Status	Pass / Fail / Ignored		
Validation Procedure	Write the procedure that confirms the validation of the requirement. (Leave empty if you ignored this particular requirement)		
Additional report	In case that you ignored this requirement for any reason, justify it.		

Table 35: R-C-25 Holistic security/privacy approach

R-C-25: Holistic security/privacy approach (Non – Functional)			
Priority	Must	**Partners responsible**	S21SEC
Rationale	Security refers to safeguarding data from dangerous threats, whereas privacy refers to the responsible use of data.		
Description	ASSIST-IoT methodologies and tools shall offer security and privacy in a wide range of users and applications, which means that sensitive data and information shall not be published or processed without permission and advanced techniques shall securely protect the data within the enablers.		
Acceptance criteria	During the execution of the pilots, the vast amount of sensitive information will be securely process if needed.		
Acceptance Status	Pending		
Validation Procedure	The Cybersecurity enablers are integrated in the projects with the pilots so only the authorized user or application will be able to access the data.		
Additional report	Not all pilots have completed their testing phase, so this requirement is left in pending state until all the pilots have completed their testing.		

Table 36: R-C-26 Optimised Security notification

R-C-26: Optimised Security notification (Non – Functional)			
Priority	Must	**Partners responsible**	S21SEC
Rationale	Increased or decreased number of notifications may lead to unidentified hazards that may reduce the efficiency of the systems.		

R-C-26: Optimised Security notification (Non – Functional)	
Description	Within the ASSIST-IoT, the number of incorrect situations that require visibility on the network shall be optimised. Depending on the deployment case of the ASSIST-IoT architecture, the number of considered events within an hour shall be refined based on the specific use case.
Acceptance criteria	During the execution of the pilots, an estimation of 10 raised alerts per hour shall guarantee the quality of the network infrastructure.
Acceptance Status	Pending
Validation Procedure	From Cybersecurity monitoring enabler will take the total cybersecurity incident number. The taken data will be from all the measurement period. and it will be evaluated by the following formula: $\frac{\textit{total number of incidents}}{(\textit{Total number of days} \times 24\ \textit{hours})}$
Additional report	Not all pilots have completed their testing phase, so this requirement is left in pending state until the pilots have completed their testing.

Table 37: R-C-27 Automated accountability

R-C-27: Automated accountability (Non – Functional)			
Priority	Must	**Partners responsible**	CERTH
Rationale	Users and/or applications communicating with a system may lead to unexpected errors. There should exist a method to diagnose such cases on the basis of reliable data.		
Description	Within ASSIST-IoT, a secure way to register interactions accurately and reliably shall be provided, utilising Distributed Ledger Technology (DLT) for keeping track of the existing interactions.		
Acceptance criteria	Critical processes shall be logged in the DLT to track the interaction within pilots		
Acceptance Status	Pass		
Validation Procedure	Several critical logs are stored in the DLT. In the context of P3b, photos that are marked as having potential damage, are hashed and stored in the DLT, in order to avoid tampering with them. In the context of P2, the alerts of workers being outbound of the permitted areas are saved, allowing to keep track of critical incidents. Finally, monitoring and notifying enabler saves metadata of occasions that gateways are found to be offline.		
Additional report	N/A		

Table 38: R-C-28 Distributed Configuration

R-C-28: Distributed Configuration (Non – Functional)			
Priority	Must	**Partners responsible**	UPV
Rationale	Distributed Configuration frameworks enables to manage the configurations of multiple enablers from a centralised location without needing to proceed in different machines		
Description	ASSIST-IoT shall allow users to configure parameters related to those software artifacts so that changes can be applied during their deployment, enhancing parametrisation and usability of ASSIST-IoT powered applications. To that end, APIs shall be used to modify their configuration, while users will proceed with modification through user-friendly interfaces.		
Acceptance criteria	Most of the developed enablers shall be fine-tuned and configured through manageability interfaces, while it shall be possible to modify more than 50 values within ASSIST-IoT.		
Acceptance Status	Pass		

R-C-28: Distributed Configuration (Non – Functional)	
Validation Procedure	To make the parameters configurable within the enablers, Helm, the packaging technology chosen, comes with great benefits. Helm allows for fine-tuned configuration using values, enabling users to modify various variables in different components of an enabler. These changes can be easily managed via the manageability enablers, using a user-friendly interface. This functionality is integrated into the Smart Orchestrator Enabler and was tested by instantiating enablers, adjusting their configurations with specific values, and monitoring the deployed resources.
Additional report	Not applicable

Table 39: R-C-29 Configurable data flows

R-C-29: Configurable data flows (Non – Functional)			
Priority	Must	**Partners responsible**	UPV
Rationale	Despite the technological advancements and intelligence provided by new technologies, stakeholders have not been focused on the usability of the technologies through user-friendly and multi-functional interfaces.		
Description	ASSIST-IoT shall be able to provide a graphical environment in which administrators may instantiate the enablers necessary for work and connect them to form a chain, or service workflow, using a Directed Acyclic Graph (DAG) interface. This DAG will allow the user to configure, select, obtain information and (most importantly) connect among and interact with enablers deployed in an ASSIST-IoT controlled environment.		
Acceptance criteria	At least 2 different topologies of enablers shall be selected and configured through the provided interface.		
Acceptance Status	Pass		
Validation Procedure	The composite services manager allows ASSIST-IoT administrators to connect previously deployed enablers to compose a composite service. Two kind of workflows have been tested: HTTP to MQTT and vice versa. In both cases, transmissions were made manually and data were distributed properly, thanks to the agents deployed by the enabler in the distributed environment when the workflows were set. Particularly, the requirement has been validated using the EDB and the LTSE as involved enablers (other targets are those from the semantic suite, not tested), with 100% transmissions correctly performed (no stress conditions applied).		
Additional report	N/A		

2.5. Performance testing

The final testing phase in ASSIST-IoT focuses on performance evaluation. This stage adopts a methodology similar to the one used for acceptance testing, with a key distinction being the selection of technical KPIs from WP8 as the primary criteria for assessing performance. The documentation in D8.3 will include similar tables for reporting, with a specific emphasis on the measurement methodology outlined in this document and the corresponding results in D8.3.

It is important to note that the scope of performance testing extends beyond these identified KPIs; they serve as a measurable baseline. The process of pilot implementation and monitoring will yield valuable insights into the performance of individual components and the overall solution. This, in turn, will introduce more mature actions. All quantitative data related to performance testing will be comprehensively compiled in D8.3. The following section presents the specific technical KPIs that have been achieved, along with documented measurement methodologies in cases where they have not been met.

2.5.1. Performance evaluation of ASSIST-IoT

2.5.1.1. KPI 4.1.1 – CPI load of GWEN

Table 40. Summary of KPI 4.1.1 CPI load of GWEN

Name	**CPU load of GWEN**				
Description	During normal operation the CPU of the GWEN must not operate at full load. Some headroom must remain for unforeseen tasks or future upgrades.				
Motivation	To validate design estimations measurements must be taken to assure CPU load during normal operation. Common design estimations are generally too low, therefore headroom must be built in.				
Initial target	75%	**Score***		**Achieved**	Yes/No
Rationale target selection	To assure lifetime and flexibility towards system changes the load shall not be higher than 75% for the whole processor (containing 4 cores and a co-processor)				
Measurement period	Last phase of pilot trials				
Partner/s responsible	Primarily NEWAYS for measurements, all others for running applications.				

Measurement methodology

Measurements will take place once the enablers and the pilot-specific software is completely deployed in the pilots. To capture the metric, this methodology will be followed:

- One-week time frame collection, per pilot in which the GWEN is deployed.
- 4 times per day a timeframe shall be measured for a duration of 15 minutes.
- This measurement shall form be analysed to give an average value of the load. The PUD enabler will be used to that end, modifying the periodicity of the metrics gathering.
- After 1 week all average values shall be combined into a grant total average which is the final KPI value.

Results and outlook

The tests have not been conducted yet – they are scheduled for M38–M39 during the last phase of pilot trials.

2.5.1.2. KPI 4.1.2 – Memory usage of GWEN processes

Table 41. Summary of KPI 4.1.2 Memory usage of GWEN processes

Name	**Memory usage of GWEN processes**				
Description	During normal operation the memory (RAM) of the GWEN must not operate at full load. Some headroom must remain for unforeseen tasks or future upgrades.				
Motivation	To validate design estimations measurements must be taken to assure memory usage during normal operation. Common design estimations are generally too low, therefore headroom must be built in.				
Initial target	75%	**Score***		**Achieved**	Yes/No
Rationale target selection	To assure lifetime and flexibility towards system changes the usage shall not be higher than 75% usage.				
Measurement period	Last phase of pilot trials				

Partner/s responsible	Primarily NEWAYS for measurements, all others for running applications.

Measurement methodology

- One-week time frame collection, per pilot in which the GWEN is deployed.
- 4 times per day a timeframe shall be measured for a duration of 15 minutes.
- This measurement shall form be analysed to give an average value of the load. The PUD enabler will be used to that end, modifying the periodicity of the metrics gathering.
- After 1 week all average values shall be combined into a grant total average which is the final KPI value.

Results and outlook

The tests have not been conducted yet – they are scheduled for M38–M39 during the last phase of pilot trials.

2.5.1.3. KPI 4.2.3 – Percentage of network connections being improved (KVI 2.2)

Table 42. Summary of KPI 4.2.3 Percentage of network connections being improved

Name	**Percentage of network connections being improved**				
Description	Auto-configurable network enabler is optimising the traffic load distribution (throughput) over the SDN network (flows re-routing) to obtain improved QoS parameters for the overall network (data losses, data transfer latency). KPI measures the percentage of connections (links) that QoS parameters were improved in the specified SDN network topology.				
Motivation	To show the advantages of traffic load optimisation in the network and QoS parameters improvement for different IoT applications.				
Initial target	20%	**Score**	32%	**Achieved**	Yes
Rationale target selection	Target was set arbitrary based on similar solutions performances.				
Measurement period	Measurement is done after traffic load changes (typically 30 sec).				
Partner/s responsible	OPL				

Measurement methodology

To measure number of improved network connections in terms of QoS parameters the continuous monitoring of all links and flows in the network is implemented using rt-sFlow for packet losses and ping based mechanisms for latency parameter. After the generated traffic load changes (different distribution scenarios) the AI based auto-configurable network mechanism optimising the flow distribution (traffic load) over the network. After changes in the routing of flows the new QoS parameters are measured. Then compare initial QoS parameters value with current one we obtain the number of the improved connections. Measuring this for different traffic load generation scenarios we can have average value of improved network connections.

Results and outlook

Initial QoS parameters for each link (data losses and latency) were measured. After the traffic load generation according to the specified scenario the new QoS parameters were measured after 30 sec. Then comparing these values, the percentage number of improved connection in average was 32% for given network load scenarios. Depends on the scenario the improvement can differ, so we calculated average value from different traffic load cases applied in the network experiments.

2.5.1.4. KPI 4.3.1 – Streaming Annotation Latency

Table 43. Summary of KPI 4.3.1 Streaming Annotation Latency

Name	**Streaming Annotation Latency**				
Description	The Semantic Annotation Enabler transforms messages that pass through its annotation channels that use a streaming broker. This KPI measures the amount of time it takes for a message to be processed, independently of the network conditions.				
Motivation	The process of annotation introduces latency, which may be important in time-sensitive IoT applications.				
Initial target	10ms	**Score***	7ms	**Achieved**	Yes
Rationale target selection	The number was chosen arbitrarily to target 100 messages per second.				
Measurement period	1/03/2023 – 30/03/2023				
Partner/s responsible	IBSPAN				

Measurement methodology

Streaming annotation is a process that depends on many factors, including network throughput, message size, and annotation configuration file size. The annotation core is a component that connects to a broker, which interfaces with clients, and passes messages to the core. To isolate the performance of the annotator itself, the message processing latency is measured. This latency is defined as the amount of time it takes to process a single message (averaged over a period of time) at the core component, without taking into account broker throughput. Even though the annotator is designed to support high volume of small messages, the latency will be tested with messages of small and medium size, to stretch the limits of the simulation. Measurements will be taken over 1 minute of constant message output from the broker, with the target of average processing latency less than 10ms per message.

Results and outlook

Message processing latency after initial was measured to be 7ms on average, with minimum of around 0,9ms and maximum of 50ms. 90% of messages are processed with latency below 10ms. It was observed, that the annotation core gets more efficient when processing a message that was already processed before, so the messages were randomised for the test.

2.5.1.5. KPI 4.3.2 – Streaming Translation Latency

Table 44. Summary of KPI 4.3.2 Streaming Translation Latency

Name	**Streaming Translation Latency**				
Description	The Semantic Translation Enabler transforms messages that pass through its annotation channels that use a streaming broker. This KPI measures the amount of time it takes for a message to be processed, independently of the network conditions.				
Motivation	The process of translation introduces latency, which may be important in time-sensitive IoT applications.				
Initial target	10ms	**Score***	10ms	**Achieved**	Yes
Rationale target selection	The number was chosen arbitrarily to target 100 messages per second.				
Measurement period	1/03/2023 – 30/03/2023				

Partner/s responsible	IBSPAN

Measurement methodology

Just like in the case of streaming annotation (see KPI.4.3.1 above), measurement of streaming translation latency was performed at the core component to isolate it from networking conditions and broker performance. Latency, defined as time needed to process a single message was measured as average over 1 minute of constant small and medium message streaming with target of less than 10 ms per message.

Results and outlook

Average message processing latency after initial tests was measured to be 10ms with minimum of 5ms and maximum of 100ms. 90% of messages are processed with latency below 35ms. Alignment and message size make a big impact on processing speed, with large alignments (1000 cells) taking significantly longer, than smaller or medium ones (<100 cells). Message and alignment size in ASSIST-IoT pilots is small, following the general principle observed in IoT, of high amount of small messages, rather than large messages that are sent less often.

2.5.1.6. KPI 4.3.3 – Streaming Annotation Clients Number

Table 45. Summary of KPI 4.3.3 Streaming Annotation Clients Number

Name	**Streaming Annotation Clients Number**				
Description	The Streaming Annotation Enabler supports parallel execution of multiple annotation channels, which may be used by multiple clients. This KPI measures the number of clients that is supported in parallel (i.e. at the same time).				
Motivation	The streaming architecture is designed to support many clients, that may join or leave, making the number variable. IoT applications often require support for multiple clients.				
Initial target	10	**Score***	20	**Achieved**	Yes
Rationale target selection	Given that each annotation channel corresponds to a kind of message semantics that should be annotated, 10 channels was deemed more than enough to cover ASSIST-IoT use cases.				
Measurement period	1/03/2023 – 30/03/2023				
Partner/s responsible	IBSPAN				

Measurement methodology

A single streaming annotator is designed to support multiple streaming channels. Performance of supported client parallelism was measured by creating a number of annotation channels, and scaling the number up, until resources on the test machine are saturated. Because the number of clients per channel is scaled on the level of the broker, each channel will have just one sender and receiver. Adding more clients per channel is supported but would make the simulation more dependent on performance of the broker, and not the streaming annotation core. A conservative target is placed at 10 parallel channels, as that is the expected maximum number of channels in a single real-world deployment, which will be realistically required.

Results and outlook

The annotation streamer is able to easily support 20 parallel channels on a desktop machine with 2 logical CPU cores. Pilot deployments on the GWEN were measured to support 10 channels without saturating the resources. Note that the amount of messages that passes though the channels was chosen not to overwhelm the machine. This means that the processing time for a message was lower than the frequency of messages. Otherwise, the internal buffer would eventually overflow. In practice, if the messages are not processed fast enough, buffer overflow becomes the main factor in resource saturation. Thus, the channel resource saturation was measured at maximum message throughput before having to use the buffer. Note, that idle channels (channels, that do not have messaging passing through them) have negligible effect on performance.

2.5.1.7. KPI 4.3.4 – Streaming Translation Clients Number

Table 46. Summary of KPI 4.3.4 Streaming Translation Clients Number

Name	**Streaming Translation Clients Number**				
Description	The Streaming Translation Enabler supports parallel execution of multiple annotation channels, which may be used by multiple clients. This KPI measures the number of clients that is supported in parallel (i.e. at the same time).				
Motivation	The streaming architecture is designed to support many clients, that may join or leave, making the number variable. IoT applications often require support for multiple clients.				
Initial target	4	**Score***	4	**Achieved**	Yes
Rationale target selection	A translation channel can support big annotation files, that represent a large chunk of semantics of an IoT ontology. Even though technically one annotation file may model an entire ontology, it is more practical and manageable to split it into multiple files. Based on the needs of ASSIST-IoT pilots, 4 parallel channels were chosen as a practical target.				
Measurement period	1/03/2023 – 30/03/2023				
Partner/s responsible	IBSPAN				

Measurement methodology

Just like the streaming annotator (see KPI.4.3.3 above), streaming semantic translator can support a number of channels in parallel, with performance concerns dependent on the same factors, as in the case of streaming annotation. Despite similar dependency on streaming brokers, the core components and internal processing infrastructure of the streaming translator and annotator are very different, so KPIs have to be measured separately. Supported client parallelism of the streaming translator was measured as maximum number of channels with one consumer and one producer per channel, before saturation of resources.

Results and outlook

The tests reveal that resource saturation is highly hardware dependent. For 4 CPU cores and configured parallelism of 4, the streaming translator easily supports 4 parallel channels that are constantly busy. In such cases network saturation occurs before local resource saturation. Constant saturation of channels is defined as a never ending stream of messages, where processing of a new message starts as soon as the previous one finishes. In practice (e.g. in testing messages in the pilots) constant saturation never occurs. Even under constant saturation, more channels may be supported. It should be noted, however, that (if allowed) the streaming translator will saturate CPU usage, and a constant CPU load will increase temperatures and, in general, prevent full efficiency in CPU performance in the long term. When not under constant saturation 32 active channels were working on 4 CPU cores. Idle channels (channels, that do not have messaging passing through them) have

2.5.1.8. KPI 4.6.2 – FL users

Table 47. Summary of KPI 4.6.2 FL users

Name	**FL users**
Description	This KPI quantifies how many simultaneous users can be involved in a common ML model training through the FL system of the project.
Motivation	This KPI aims at demonstrating the scalability capabilities of the ASSIST-IoT FL system

Initial target	10	**Score***		**Achieved**	Yes/No
Rationale target selection	The target of 10 simultaneous users/clients involved in the FL system was agreed internally, channels, as that is the expected maximum number of users in a single real-world deployment, that will be realistically required.				
Measurement period	Several tests have been performed along the FL development stages. As long as new features were incorporated, they were tested with at least 10 clients involved in the training.				
Partner/s responsible	PRO				

Measurement methodology

A minimum of 10 available clients are set up in the initial configuration fields from the GUI that connects with the FL Orchestrator. Then as long as the number of connected clients (i.e., websocket clients with their IP addresses and ports) are below 10, the FL Orchestrator enforce to stop the training and the tests were considered a failure.

Results and outlook

As it can be seen in the below screenshots, 10 FL LOs were deployed in the K8s cluster under test. In addition, the bottom screenshot presents FL Orchestrator websocket server logs, in which its messages are properly received at the different websocket clients, which in turn correspond to the 10 FL LOs.

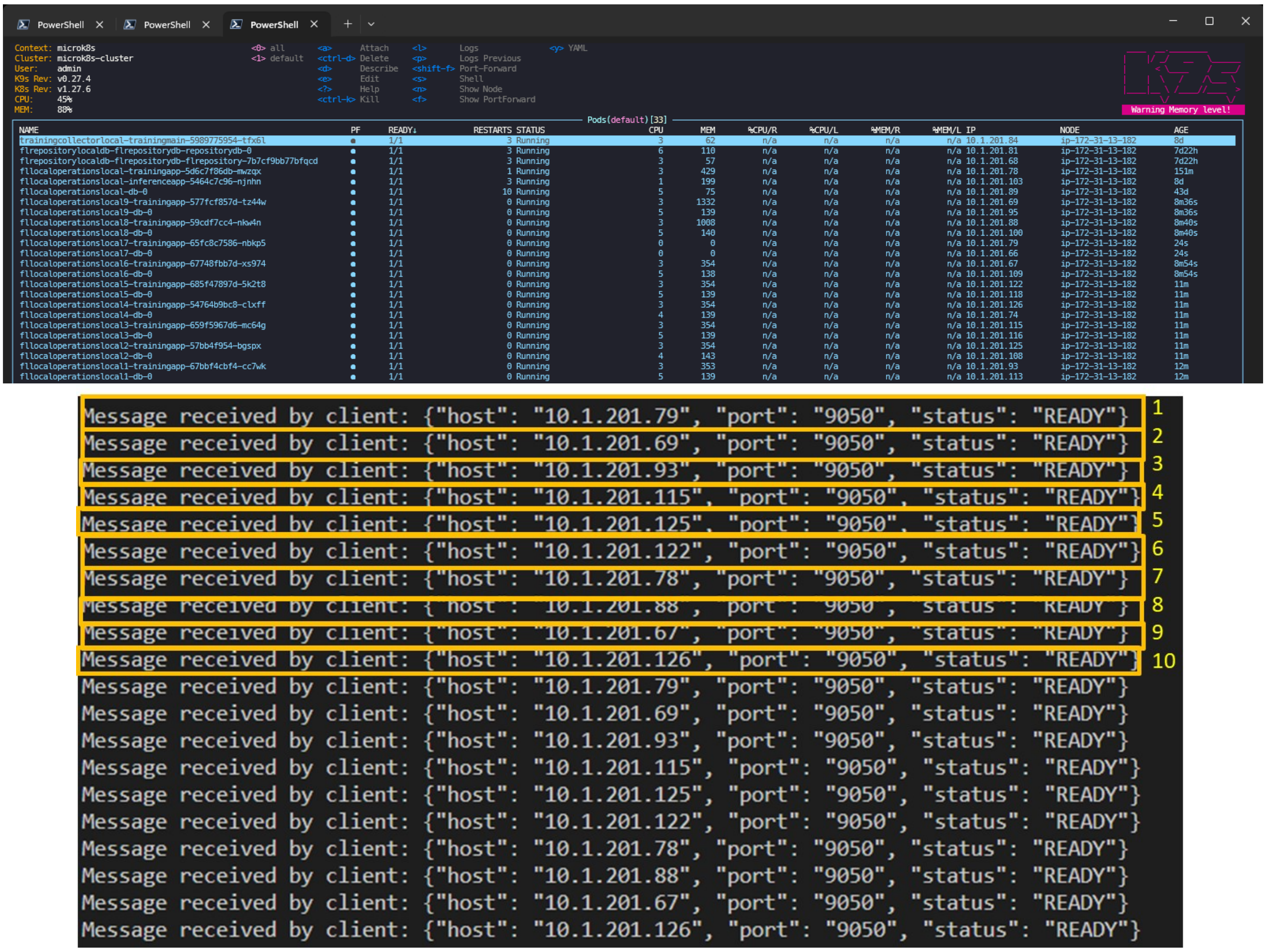


Figure 3. FL LOs running in the K8s cluster (top). FL Orchestrator WebSocket logs illustrating the propper communication with the WebSocket client IPs from the 10 FL LOs (bottom).

Finally, the following figure presents the GUI of the FL Orchestrator informing the successful execution of the FL training with the above 10 FL Local Operations.

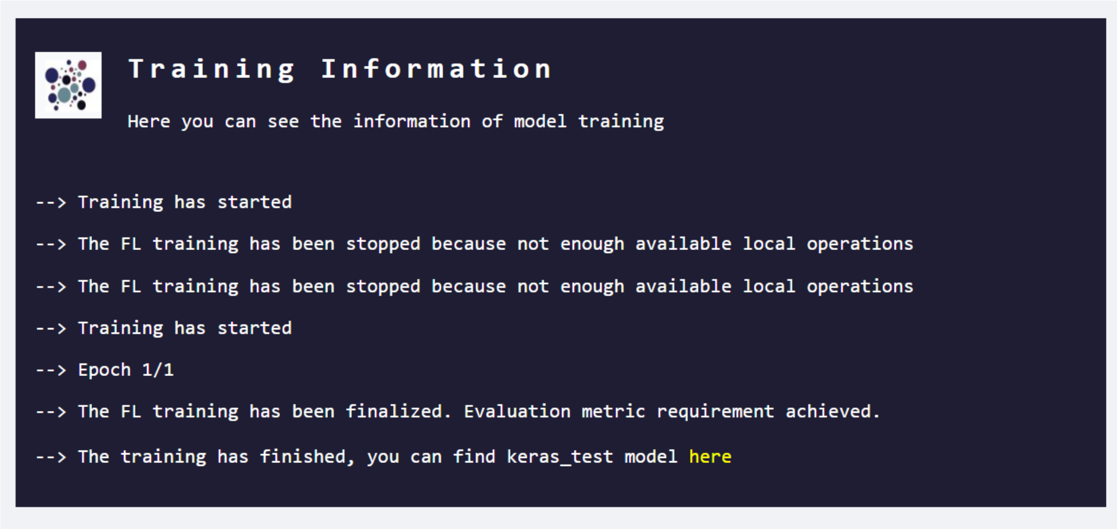


Figure 4. GUI messages informing about the FL training with the 10 FL LOs under test.

2.5.2. Performance evaluation of pilot results

2.5.2.1. KPI 2.1.5 – Remote wireless bandwidth

Table 48. Summary of KPI 2.1.5 Remote wireless bandwidth

Name	**Remote wireless bandwidth**				
Description	This KPI evaluates the transmission bandwidth of Malta Freeport terminal wireless access.				
Motivation	The use of high bandwidth systems will help remote crane operators to properly visualize the video streams captured by the cameras, and, consequently, helping the remote operating conditions.				
Initial target	> 60 Mbps	**Score***		**Achieved**	Yes/No
Rationale target selection	This value was obtained from previous remote operating cranes conditions by Konecranes in other container terminals.				
Measurement period	10 times from the ROS cranes were ready for operation				
Partner/s responsible	TL				

Measurement methodology

To evaluate this KPI iperf3 open-source tool is used. Iperf3 is a cross-platform CLI-based program that performs real-time network throughput measurements. It is installed on the two RTG cranes remote management system.

Then, in order to test the network throughput, the following commands are performed:

1. Connect a remote machine as the server and fire up iperf3 in server mode using `-s` flag, and listening on port 5201 by default. The size format to be reported can be specified (k/K → Kbits/Kbytes; m/M → Mbits/MBytes; g/G → Gbits/GBytesGbytes) using the `-f` switch.
2. Then on the machine to be treated as the client (and where the actual benchmarking takes place), client mode using `-c` flag is specified, plus the server IP (or domain or hostname).

3. After about 18 to 20 seconds, the client terminates the communication benchmark and produce results indicating the average throughput for the benchmark.

The bandwidth measurement was conducted for 10 different times with different weather conditions (sunny, cloudy, rainy, daylight, nightlight).

Results and outlook

Table 49. Result of KPI 2.1.5 Remote wireless bandwidth

Times measured	Mean	Unit	> 60 Gbps
Day 1		Mbps	
Day 1		Mbps	
Day 2		Mbps	
Day 3		Mbps	
Day 4		Mbps	
Day 5		Mbps	
Day 6		Mbps	
Day 7		Mbps	
Day 8		Mbps	
Day 9		Gbps	
Day 10		Gbps	

2.5.2.2. KPI 2.1.6 – Remote wireless latency

Table 50. Summary of KPI 2.1.6 Remote wireless latency

Name	**Remote wireless latency**				
Description	This KPI evaluates the application latency from the source (camera) to the destination (screen) in the remote operating cranes system.				
Motivation	The remote operation use of the RTG cranes to be deployed requires of ultra-low latencies to support the tactile internet capabilities needed to provide a successful user experience for those crane drivers that are remotely managing the crane.				
Initial target	< 20 ms	**Score***		**Achieved**	Yes/No
Rationale target selection	This value was obtained from previous remote operating cranes conditions by Konecranes in other container terminals.				
Measurement period	10 times from the ROS cranes were ready for operation				
Partner/s responsible	TL				

Measurement methodology

For testing the latency of the remote operating system of RTG crane the basic OS-native mechanisms will be used: **ping** and **traceroute**/**tracert**. While **ping** is the simplest mechanism that measures the Round-Trip Time between a client and a specified target server, **traceroute** uses the Time To Live field of IP packets to discover intermediate routers between a source and a destination, providing more insights into the origin of network latency problems. In this KPI, **ping** has been the native command used in the following way:

1. A console or terminal window is opened, and just by typing ping domain, it provides the RTT latency of 4 IP packets to the specified IP address destination.

Like the previous KPI, the latency measurement was conducted for 10 different times with different weather conditions (sunny, cloudy, rainy, daylight, daynight).

Results and outlook

Table 51. Result of KPI 2.1.6 Remote wireless latency

Times measured	Mean	Unit	< 20 ms
Day 1		ms	
Day 1		ms	
Day 2		ms	
Day 3		ms	
Day 4		ms	
Day 5		ms	
Day 6		ms	
Day 7		ms	
Day 8		ms	
Day 9		ms	
Day 10		ms	

2.5.2.3. KPI 2.2.9 – Worker alert latency

Table 52. Summary of KPI 2.2.9 Worker alert latency

Name	**Worker alert latency**				
Description	The delay between the worker entering a danger zone and them being alerted should be lower than the specified target.				
Motivation	A large delay in the notification would prevent the system from effectively warning workers of dangers on the construction site,				
Initial target	1.5 s	**Score***		**Achieved**	Yes/No
Rationale target selection	A delay of 1.5 seconds gives the worker enough time to react when entering a danger zone that is typically a few meters wide.				
Measurement period	After the main trials on the construction site.				
Partner/s responsible	IBSPAN				

Measurement methodology

The scenario will be simulated by instructing a worker to enter a temporarily set dangerous zone and then measuring the time between the incursion and the triggering of the worker's alarm system. There will be no actual risk to the worker during the test, to ensure their safety. The test will be supported by additional painted lines, to delimit the dangerous zone set in the BIM model. The delay will be measured by examining a recorded video of the experiment with the wristband clearly visible, to accurately assess the time at which the notification arrived on the device.

The experiment will be repeated several times in various scenarios (different walking directions, danger zone orientations, etc.).

Results and outlook

The tests were not conducted yet – they are scheduled for M37–M38.

2.5.2.4. KPI 2.2.10 – OSH manager notification latency

Table 53. Summary of KPI 2.2.10 OSH manager notification latency

Name	**OSH manager notification latency**				
Description	The latency between the worker's incursion into a danger zone and the manager's notification should be minimized.				
Motivation	The latency should be minimized so that the OSH manager can react in time to the dangerous situation.				
Initial target	5 s	**Score***		**Achieved**	Yes/No
Rationale target selection	A delay of 5 seconds was deemed both acceptable from the OSH standpoint (in comparison to the current practice it is nearly instantaneous) and achievable for the system, bearing in mind the various delays introduced by the network and all involved software components.				
Measurement period	After the main trials on the construction site.				
Partner/s responsible	IBSPAN				

Measurement methodology

The measurement setup will be the same as in KPI 2.2.9. The screen of the OSH manager's device will be recorded using an additional camera, synchronized with the one observing the worker. The delay will be measured by examining the recorded footage.

Results and outlook

The tests were not conducted yet – they are scheduled for M37–M38.

2.5.2.5. KPI 2.3.7 – Sever capacity to manage and monitor vehicle fleet

Table 54. Summary of KPI 2.3.7 Sever capacity to manage and monitor vehicle fleet

Name	**Sever capacity to manage and monitor vehicle fleet**				
Description	The amount of vehicles in a vehicle fleet is likely the most challenging number of GWENs within all ASSIST-IoT Pilot projects. Therefore, a KPI needs to be implemented which reflects the server capacity to handle a large number of edge nodes.				
Motivation	Even single model vehicle fleets can consist of a large amount of vehicles, as certain models are produced in numbers of several 100.000 units per year. Therefore, it is essential, that the ASSIST-IoT infrastructure is able to handle this vast amount of edge nodes. This can be supported by intelligent solutions like load balancing and queuing mechanisms.				
Initial target	≥200 -> > 10.000	**Score***	90.000	**Achieved**	Yes
Rationale target selection	The first target of ≥200 edge node was selected in initial phases, considering the performance of the first versions of the orchestrator. However, a specific version of the				

	enabler has been developed to smoothly enable the management of large number of edge nodes, so now the minimal internal target considered has been set to 10.000.
Measurement period	During the third phase of the trials.
Partner/s responsible	UPV

Measurement methodology

The methodology consists in stressing the system to ensure that the enablers involved (EDBE, Fault-tolerance data enabler, Smart orchestrator – group-based flavour) are capable of handling the pilot 3a architecture. The steps considered are the following:

1. The cloud setup is put into place. This entails deploying and connecting the aforementioned enablers with the orchestrator, using the flavour for the pilot.
2. An MQTT traffic simulator is used to create devices and subscribe them to the topics of the MQTT topics used for managing the fleet. These are connected and data traffic is sent simulating real conditions (i.e., time span, data).
3. While number of clients grow (they are created in a stepwise way, not all at once), it is considered that the system supports them if data simulated arrives correctly to the cloud site of pilot 3a.
4. The total number of edge nodes supported will be that before a system fail, or error rate of transmitted data is above 0.5%.

Results and outlook

At this moment, a total of 90.000 simulated edge clients have been simulated and supported. Errors came up not due to the performance of the enablers, but of the hardware and virtualized environments, as the network cards used were not able to support the high volume of traffic generated.

The KPI is considered fulfilled. In future tests, it would be convenient to repeat it considering bonding techniques of the Ethernet adapters, so overall throughput supported is multiplied.

2.5.2.6. KPI 2.4.5 – Deliver vehicle images to the user in time, after vehicle scanning

Table 55. Summary of KPI 2.4.5 Delivery time of scanned images

Name	**Delivery time of scanned images**				
Description	The KPI is the time between the moment of the end of the vehicle scan till the time of delivery of the first scanned vehicle images for the potential review by the end-user of the system (vehicle inspector). This time shall be part of the Business KPIs kept in the corresponding enabler				
Motivation	The digital vehicle scanning is a substantial part of the digitalisation process of the vehicle exterior conditions in various business cases in the automotive industry. Taking the example of the automotive branch and passenger car services this digitalisation should increase the corresponding process productivity. Todays' manual reviewing process takes about 15-20 min to go outside the office buildings to inspect the car and return for the customer consulting process for the actual service contract. Having the vehicle images already on the screen of the customer consultant (vehicle inspector) and doing the reviewing remotely the branch process can save a lot of time and have increased productivity.				
Initial target	3-5 min	**Score***		**Achieved**	Yes/No

Rationale target selection	The expected delivery time of the vehicle scanned images to the customer consultant must be in the range of 3 to 5 min. Otherwise there will be a collision within the customer service procedure steps and will not deliver the badly needed productivity enhancement.
Measurement period	-
Partner/s responsible	TwoTronic

Measurement methodology

Having the time of scanning and the time of readiness to deliver any requested picture of the scanned ones we can measure the delivery time. The scanning time is recorded by the digital scanner system and is written into the meta-data of the corresponding scan. They are saved into the SQL-database of the LTSE and can be used for the measurement. Also the (global) time of the arrival of the scanned image sets of the scans are available within the ASSIST-IoT system software at both edge and cloud sites. An active notification service is executed after the scan delivery to both the ASSIST-IoT-Edge-PC and the cloud system. The Business KPI enabler has access with the cloud system to the corresponding enablers and can calculate and keep this KPI for the cloud part of the application, i.e. where the user access to the scans is configured by the cloud. As this is usually longer than the local delivery time on the edge, we can evaluate this KPI on the safe side.

Results and outlook

The tests were not conducted yet – they are scheduled for M37–M38.

3. Packaging and releasing

Relevant information about T6.3 task – packaging and releasing, can be consulted in D6.7. The summary of that deliverable is the following [2.]: *"Regarding* ***packaging****, Helm charts have been used for packaging the enablers of the project, making some* ***adaptations*** *over their typical structure and conventions to ease enablers integration in the project. This report delves on these adaptations, presenting the final version of the chart generation to smooth their preparation (and, to adapt existing ones). With respect to* ***releasing****, the artifacts registries and licensing strategy (leveraging Apache 2.0 or similar permissive licenses) are also updated. Until M30, code, container images and chart packages have been kept primarily on private repositories; however, from now on,* ***container images and packages*** *will be hosted in the project's* ***public*** *registries, to be released in GitHub, DockerHub and ArtifactHub repositories once the technical developments are finished".*

In this deliverable, complementary to the former, the final summary of licenses is presented, as well as the structure of public repositories that are being used for long-term hosting of the enablers' code, container images and Helm charts.

3.1. Enablers' packaging status and licenses

All enablers should include a LICENSE file (without extension or .txt) in the root of the project of the enabler, with the license applied to it.

Also, all enablers should include a NOTICE file (without extension or .txt) at the root of the enabler, and fill it with the following content:

- Acronym and title of the project, and funding acknowledgement;
- Name of the enabler;
- Copyright of the developer companies that contributed to the enabler;
- Included software (if any): source code you have included, meaning in copied and pasted from its origin, or modified (if allowed by its license);
- Used software: list other code you have used and linked, meaning, the libraries used. Place a link to the software's repository and license they have;
- License summary: list of all the licenses of the code included and used above. An example of the Traffic classification enabler is depicted below:

```
ASSIST-IoT - Architecture for Scalable, Self-*, human-centric, Intelligent, Se-
cure, and Tactile next generation IoT

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 957258.

Traffic classification enabler

Copyright 2020-2023 Universitat Politècnica de València

I. Included Software
   -  Deep-Packet (https://github.com/munhouiani/Deep-Packet), MIT license

II. Used Software

   -  colorama 0.4.4 (https://github.com/tartley/colorama/tree/0.4.4), BSD-3-
      Clause license
   -  joblib (https://github.com/joblib/joblib), BSD-3-Clause license
   -  Flask 2.0.2 (https://github.com/pallets/flask/tree/2.0.x), BSD-3-Clause li-
      cense
```

- itsdangerous 2.0.1 (https://github.com/pallets/itsdangerous/tree/2.0.1) BSD-3-Clause license
- Jinja2 3.0.3 (https://github.com/pallets/jinja/tree/3.0.3), BSD-3-Clause license
- MarkupSafe 2.0.1 (https://github.com/pallets/markupsafe/tree/2.0.1), BSD-3-Clause license
- Werkzeug 2.0.2 (https://github.com/pallets/werkzeug/tree/2.0.x), BSD-3-Clause license
- Gunicorn 20.1.0 (https://github.com/benoitc/gunicorn/tree/20.x), custom license (see list below)
- Requests 2.27.1 (https://github.com/psf/requests/tree/v2.27.x), Apache-2.0 license
- flask_wtf 1.0.0 (https://github.com/wtforms/flask-wtf/tree/1.0.x), BSD-3-Clause license
- peewee 3.14.10 (https://github.com/coleifer/peewee/tree/3.14.10), MIT license
- wtforms (https://github.com/wtforms/wtforms), BSD-3-Clause license
- pymysql (https://github.com/PyMySQL/PyMySQL), MIT license
- click 8.1.3 (https://github.com/pallets/click/tree/8.1.x), BSD-3-Clause license
- jupyterlab 3.4.7 (https://jupyter.org/governance/projectlicense.html), BSD-3-Clause license
- matplotlib 3.5.3 (https://matplotlib.org/stable/users/project/license.html), custom license (PSF-based, see list below)
- datasets 2.5.1 (https://github.com/huggingface/datasets/tree/2.5.1), Apache-2.0 license
- pandas 1.4.4 (https://github.com/pandas-dev/pandas/tree/1.4.x), BSD-3-Clause license
- plotly 5.10.0 (https://github.com/plotly/plotly.py/tree/v5.10.0), MIT license
- pyspark 3.3.0 (https://github.com/apache/spark/tree/master/python/pyspark), Apache-2.0 license
- pytorch-lightning 1.7.7 (https://github.com/Lightning-AI/lightning/tree/1.7.7), Apache-2.0 license
- scapy[complete] 2.5.0rc1 (https://github.com/secdev/scapy/tree/v2.5.0rc1), GPL-2.0 license
- scikit-learn 1.1.2 (https://github.com/scikit-learn/scikit-learn/tree/1.1.X), BSD-3-Clause license
- seaborn 0.11.2 (https://github.com/mwaskom/seaborn/tree/v0.11.2), BSD-3-Clause license
- tensorboard 2.10.0 (https://github.com/tensorflow/tensorboard/tree/2.10), Apache-2.0 license

III. List of licenses

- BSD-3-Clause license
- Gunicorn license
- Apache-2.0 license
- MIT license
- matplotlib license
- GPL-2.0 license

The licensing considered for each enabler is shown in the following table, depicting as well the current packaging status (in green background, encapsulated and packaged with Helm chart; in yellow, encapsulation exception - see D3.7 [6.], and in grey, hardware design).

Table 56: Software licensing of enablers

Owner(s)	Enablers	Access rights	Protection	License Link	Description for private license
UPV	*Smart Orchestrator*	*Permissive, Open Source*	*Apache 2.0*	[*Apache 2.0*]	-
	Traffic Classification				
	Multi-link				
	SD-WAN				
	WAN Acceleration				
	VPN				
	Resource Provisioning				
	Enablers Manager				
	Composite Services manager				
	Clusters and Topology manager				
SRIPAS	*Sematic Repository*	*Permissive, Open Source*	*Apache 2.0*	[*Apache 2.0*]	-
	Semantic Translation				
	Semantic Annotation				
	Location Processing				
	Automated Configuration				
	FL Training Collector				
	FL Repository				
	FL Local Operations				
PRO	*LTSE*	*Permissive, Open Source*	*Apache 2.0*	[*Apache 2.0*]	-
	Tactile Dashboard				
	Business KPI				
	Video Augmentation				
	FL Orchestrator				
	Self-healing device	*Copyleft*	*GNU AGPLv3*	[*GNU AGPLv3*]	
CERTH	*OpenAPI management*	*Copyright, Proprietary Licence*	*Proprietary license*	-	*CERTH's policy is to retain ownership of its products. For commercial*
	Monitoring and Notifying				
	Logging and auditing				
	Data Integrity verification				

Owner(s)	Enablers	Access rights	Protection	License Link	Description for private license
	Distributed broker				*use a private agreement should be granted.*
	DLT-based FL				
S21	*Authorisation*	*Proprietary*	*Copyright, Proprietary Licence*	-	-
	Identity manager	*Permissive, Open Source*	*Apache 2.0*	[*Apache 2.0*]	
	Cybersecurity Monitoring Agent				
	Cybersecurity Monitoring		*Incident Detection: Apache 2.0 Incident Response: GNU AGPLv3*	*Incident Detection; Apache 2.0 Incident Response: GNU AGPLv3*	
ICCS	*Edge data broker*	*Permissive, Open Source*	*Apache 2.0*	*Apache 2.0*	-
	PUD				
	MR	*Copyleft*	*GNU AGPLv3*	*GNU AGPLv3*	
OPL	*SDN Controller*	*Permissive, Open Source*	*Apache 2.0*	*Apache 2.0*	
	Auto Configurable Network				
NEWAYS	*GWEN*	*Proprietary*	-	-	-
	UWB Localization tracking				

Table 57: Third party IPR compliance

Enabler	Component	3d party OSS dependency	License	License Type	Link
IdM	*Identity manager*	*Keycloak*	*Apache 2.0*	*Permissive*	https://github.com/keycloak/keycloak
Cybersecurity Monitoring	*Decoder, rule engine, correlator*	*Wazuh*			https://github.com/wazuh/wazuh
Cybersecurity Monitoring	*Incident response*	*TheHive*	*GNU AGPLv3*	*Permissive*	https://github.com/TheHive-Project/TheHive
DLT Enablers	*ALL*	*Hyperledger Fabric*	*Apache 2.0*	*Permissive*	https://github.com/hyperledger/fabric
Traffic classification enabler	*ALL except the Traffic classification API*	*Deep traffic*	*MIT license*	*Permissive*	https://github.com/munhouiani/Deep-Packet/tree/master
Semantic Repository	*File storage*	*MinIO*	*AGPL v3*	*Copyleft*	https://min.io/compliance
Semantic Repository	*Database*	*MongoDB*	*SSPL*	*Source-available*	https://www.mongodb.com/legal/licensing/community-edition
Semantic Annotation	*Configuration persistence*				
FL Repository	*Database*				
Monitoring and Notifying	*Database*				

Enabler	Component	3d party OSS dependency	License	License Type	Link
Semantic Translation	*Storage*	*PostgreSQL*	*PostgreSQL License*	*Permissive*	https://www.postgresql.org/about/licence/
Location Processing	*Database*				
Long Term Storage	*SQL server*				
OpenAPI management	*Publisher*				
OpenAPI management	*Gateway*	*Kong*	*Apache 2.0*	*Permissive*	https://github.com/Kong/kong
OpenAPI management	*Portal*	*Konga*	*MIT license*	*Permissive*	https://github.com/pantsel/konga
Automated Configuration	*Eventstore*	*EventStore DB*	*3-clause BSD*	*Permissive*	https://github.com/EventStore/EventStore/blob/master/LICENSE.md
Smart Orchestrator	*Scheduler*	*Mck8s*	*Apache 2.0*	*Permissive*	https://github.com/moule3053/mck8s
Smart Orchestrator	*Smart orchestrator microservices*	*Helm*			https://github.com/helm/helm
Smart Orchestrator	*Metrics-Server*	*Prometheus Federate*			https://github.com/prometheus/prometheus
Smart Orchestrator	*Networking component*	*Cilium*			https://github.com/cilium/cilium
Smart Orchestrator	*Networking component*	*Flannel*			https://github.com/flannel-io/flannel
Smart Orchestrator	*OSM*	*OSM (until v4.0.0)*			https://osm.etsi.org/gitlab/osm
Smart Orchestrator	*Scheduler*	*Neural Prophet*	*MIT*	*Permissive*	https://github.com/ourownstory/neural_prophet
Resource Provisioning	*Training module*				
SD-WAN	*CRD controller, CNF*	*icn-sdwan*	*Apache 2.0*	*Permissive*	https://github.com/akraino-edge-stack/icn-sdwan
WAN Acceleration	*Central controller*				
Multi-link	*Client / Server VPN*	*OpenVPN*	*GNU/GPL v2*	*Permissive*	https://github.com/OpenVPN/openvpn
PUD	*Server, Dashboards*	*Prometheus*	*Apache 2.0*	*Permissive*	https://github.com/prometheus/prometheus
BKPI	*UI, Server, plugins*	*Kibana*	*ELv2 or SSPL*	*Permissive*	https://www.elastic.co/es/licensing/elastic-license
LTSE	*LTSE NoSQL cluster*	*Elastic*			
Video augmentation enabler	*ML_trainer, Inference engine*	*Tensorflow*	*Apache 2.0*	*Permissive*	https://github.com/tensorflow/tensorflow
EDBE	*MQTT Broker*	*VerneMQ*	*Apache 2.0*	*Permissive*	https://vernemq.com/vernemq-licenses.html

3.2. Artifact repositories setup

The Consortium is following the releasing plan depicted in D6.7 (see Figure 5). Up to now, enablers' code has been kept in a private GitLab repository, following the internal structure defined in D6.1:

- WPx/ (main folder, either WP4 or WP5, with WP leader as owner).
    - Txx/ (task-level folder, T4.1 – T4.4 / T5.1 – T5.5, with Task leaders as owners).
        - Enabler_x (project for the enabler, with enabler responsible as owner) .

Each enabler had its own project, with its own container and package registries for containers and charts, respectively. To facilitate the use for external users, a public registry was made available for containers and Helm charts, with the documentation available in the ReadTheDocs page (see Section 4).

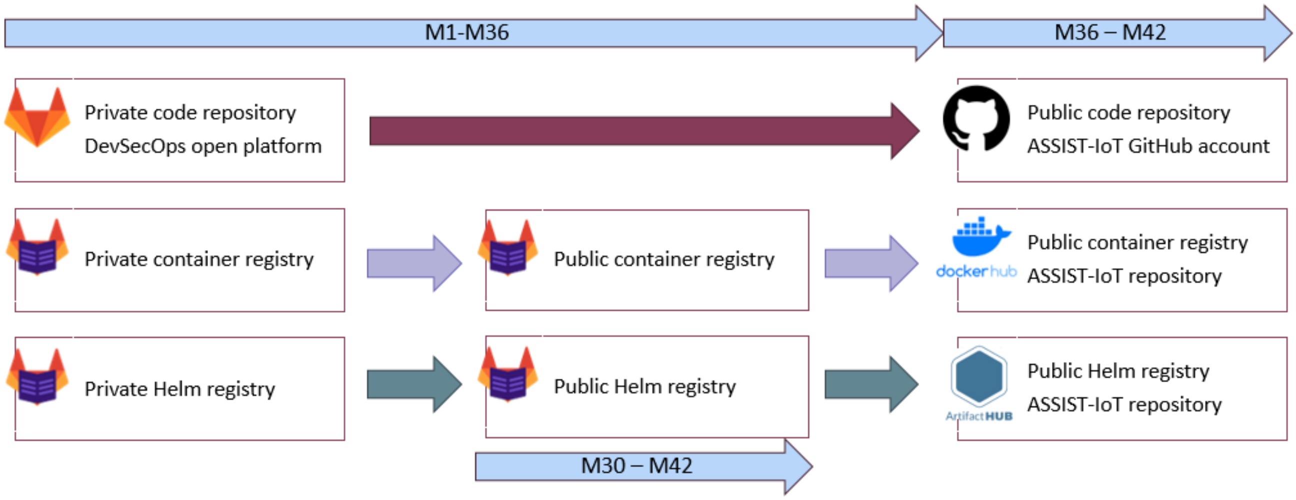


Figure 5. Artifacts registries

Currently, all the artifacts – code, container images and Helm charts, are being or have been moved to public repositories. Specifically:

- An **ASSIST-IoT profile** has been created in **GitHub**, with 40 repositories created – one per enabler. Each repository is managed by the enabler owner, open to the rest of the Consortium and external adopters to make pull requests and publish issues.
- Additionally, each enabler repository is linked with **ArtifactHUB registry**, easing their management and the setup of end-to-end DevSecOps pipelines. It is important to remark that project activities have been carried out in the private GitLab, and developed pipelines will not be shifted to GitHub.
- Finally, container images are published in **DockerHub**, in a community named ASSIST-IoT. In this case, images follow this naming convention: assist-iot/{enabler-name}_{component-name}:{tag}.

4. Technical support and documentation

This deliverable marks the final state of Technical and Support documentation, presenting the most up-to-date and comprehensive versions of each Enabler. Additionally, it provides links to the final wikis for each enabler. The installation prerequisites and deployment procedures for the ASSIST-IoT environment remain consistent, and they are available in D6.6 [3.]. In particular, the wikis featured in this deliverable offer insight into the maturity level and developments of the enablers up to M36, displaying their finalized state while incorporating the feedback received from Open Callers (round 1st) through the questionnaire. Alongside the wiki of the project now that we are in the final steps there is also available open source code for most of the enablers and corresponding Docker images these can be found in the repositories of the project as shown in the following links.

Material	Link
Technical Documentation	https://assist-iot-enablers-documentation.readthedocs.io/en/latest/index.html
Open-Source code	https://github.com/assist-iot
Docker images	https://hub.docker.com/repositories/assistiot

4.1. Documentation Progress Report

4.1.1. Horizontal Planes' Enablers

4.1.1.1. Device and Edge Plane

4.1.1.1.1. Smart Devices

Enabler: *Localization Tag*	**Id:** *T41E1*
Owner and Support: *NEWAYS, Pilot Stakeholders (MOSTOSTAL, FORD)*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Ongoing*	
Enabler Description	
The ASSIST-IoT localisation tag is a Smart IoT device used for people's localisation purposes, especially in indoor cases. This device has tag functionality, and it contains a buzzer and red LED. The buzzer is used to indicate to the person that he/she is in a restricted area. The button is used to alert the system when the worker detects an accident and immediate help is needed.	
Keywords / Key components	
UWB, low power, indoor localisation	
Link for wiki	
Wiki: https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/device/localization_tag.html	

Used by Open Callers, suggestions and comments
Open Caller name: Hopcast
Suggestion and comments: N/A

4.1.1.1.2. GWEN

Enabler: GWEN	**Id:** *T41E3*
Owner and Support: *NEWAYS, Pilot Stakeholders (MOSTOSTAL, FORD)*	
Related Deliverable/s: *D3.5, D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
The GateWay/EdgeNode (GWEN) is a device used as an edge computing interface between sensors & actuators and or systems on one side and a communication network on the other side. Sensors actuators and systems can be connected through wired and/or wireless interfaces. The interface with a network can also be wired or wireless. *Available wired interfaces are: Ethernet, CAN & CAN FD, USB2 and USB3, PCIe, HDMI, Camera interface (CSI) SD card.* *Available wireless interfaces are: WiFi, Bluetooth and 3G/4G/5G. In addition, an UWB interfaces via USB is available for localisation purposes.* *The GWEN also contains computational power to be able to operate AI algorithm, while Docker in combination with Kubernetes is used as container runtime on top of Linux as OS, together with several apps for specific purposes within pilots.*	
Keywords / Key components	
Ethernet, WIFI, Bluetooth, 5G, UWB, CAN, USB, Docker, Linux	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/device/edge_node.html	
Used by Open Callers, suggestions and comments	
Open Caller name: N/A	
Suggestion and comments: N/A	

4.1.1.2. Smart Network and Control Plane

4.1.1.2.1. Smart Orchestrator

Enabler: *Smart Orchestrator*	**Id:** *T42E1*
Owner and Support: *UPV*	

Related Deliverable/s: *D4.1, D4.2, D4.3*
Status of Enabler: *Completed*
Enabler Description
This enabler controls the whole lifecycle of Containerised Functions, network and not-network related, from their instantiation to their termination, allowing their deployment in any K8scluster available. It also controls the underlying network, implementing specific network policies to allow/deny traffic according to ASSIST-IoT networking specifications.
Keywords / Key components
Orchestrator, Helm, Microservices, Intent, Kubernetes
Link for wiki
Wiki:[*https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/smart/smart_orchestrator.html*](https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/smart/smart_orchestrator.html)
Used by Open Callers, suggestions and comments
Open Caller name: HopCast, CHEaaS **Suggestion and comments: Well-structured Technical Documentation**

4.1.1.2.2. SDN Controller

Enabler: *SDN controller*	**Id:** *T42E2*
Owner and Support: *UPV*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
The SDN Controller is the key element of an SDN-enabled network, where the main functionalities are related to network management, operation and maintenance, allowing topology management, network configuration, network control and network operations, among other features. Two solutions are investigated based on open source implementation: µONOS and Tungsten.	
Keywords / Key components	
SDN, network configuration, network management, network monitoring, topology	
Links for wiki	
Wiki:[*https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/smart/sdn_controller.html*](https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/smart/sdn_controller.html) **Dockerhub:**	

Artifact Repository:
Used by Open Callers, suggestions and comments
Open Caller name: N/A **Suggestion and comments:** N/A

4.1.1.2.3. Auto-configurable network enabler

Enabler: *Auto-configurable network*	**Id:** *T42E3*
Owner and Support: *OPL*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler provides solution for network configuration using the SDN Controller of an ASSIST-IoT ecosystem. The policy based solution using the northbound APIs of the SDN Controllers that improves the performance of selected KPIs of the network (required by use case applications). The strategies are under specification based on requirements of network performance and quality for use cases applications. Solution for network resources optimisation are under investigation.	
Keywords / Key components	
Network configuration, policy based management, KPI, network performance, network quality	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/smart/auto_configurable_network_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: N/A **Suggestion and comments:** N/A	

4.1.1.2.4. Traffic classification enabler

Enabler: *Traffic classification*	**Id:** *T42E4*
Owner and Support: *UPV*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	

The aim of this enabler is to classify network traffic into a number of application classes (video streaming, VoIP, Network control, best effort, OAM, etc.), making use of an AI/ML framework and dedicated algorithms. The traffic classification enabler can be seen as a service of the application layer of the general SDN architecture.
Keywords / Key components
Network Traffic, Classifier, AI/ML
Link for wiki
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/smart/traffic_classification_enabler.html
Used by Open Callers, suggestions and comments
Open Caller name: N/A **Suggestion and comments: N/A**

4.1.1.2.5. Multi-link enabler

Enabler: *Multi-link*	**Id:** *T42E5*
Owner and Support: *UPV, PRO, TL*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
Multi-link wireless network capabilities provide the possibility of sending IP-based data over different Radio Access Networks and different channels in each of them (for instance, regarding cellular, using more than 1 connection). Besides, it should provide reliability and redundancy mechanisms: in case one channel is down, signal cannot be lost or at least it should be recovered almost in real time (to achieve it, data will be sent via more than one wireless network).	
Keywords / Key components	
Reliability, Redundancy, Connectivity	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/smart/multi_link_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: PROUD5G **Suggestion and comments: Deeper code tests before submitting changes to GitLab**	

4.1.1.2.6. SD-WAN enabler

Enabler: *SD-WAN*	**Id:** *T42E6*
Owner and Support: *UPV*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
The objective of this enabler is to provide access between nodes from different sites based on SD-WAN technology. In particular, this enabler will implement mechanisms to connect K8s clusters via private tunnels, facilitating (i) the deployment and chaining of virtual functions to secure connections between them and/or towards the Internet and (ii) the implementation of functions to optimise WAN traffic.	
Keywords / Key components	
SD-WAN, Controller, Connectivity	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/smart/sd_wan_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: N/A **Suggestion and comments: N/A**	

4.1.1.2.7. WAN Acceleration enabler

Enabler: *WAN acceleration*	**Id:** *T42E7*
Owner and Support: *UPV*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler aims at increasing the efficiency of data transfer in Wide Area Network. This enabler will contain a set of independent, standalone CNFs with that purpose. These functions can be either chained (so data that requires of different techniques travels through the different functions) or selected for specific purposes.	
Keywords / Key components	
Traffic shaping, Compression, Traffic optimisation	

Link for wiki
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/smart/wan_acceleration_enabler.html
Used by Open Callers, suggestions and comments
Open Caller name: N/A **Suggestion and comments: N/A**

4.1.1.2.8. VPN enabler

Enabler: *VPN*	**Id:** *T42E8*
Owner and Support: *UPV*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler facilitates the access to a node or device from a different network to the site's private network using a public network (e.g., the Internet) or a non-trusted private network, by establishing a dedicated encrypted tunnel.	
Keywords / Key components	
VPN, Tunnelling, Connectivity	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/smart/vpn_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: ATHEMS **Suggestion and comments: N/A**	

4.1.1.3. Data management Plane

4.1.1.3.1. Semantic Repository enabler

Enabler: *Semantic Repository*	**Id:** *T43E1*
Owner and Support: *SRIPAS, MOW, PRODEVELOP, KONECRANES, FORD*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	

Enabler Description
This enabler offers a "Nexus" for data models and ontologies, that can be uploaded in different file formats, and served to users with relevant documentation. This enabler is aimed to support files that describe data models or support data transformations, such as ontologies, schema files, semantic alignment files etc.
Keywords / Key components
Data Models, Ontologies, Repository
Link for wiki
Wiki:[*https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/datamanagement/semantic_repository_enabler.html*](https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/datamanagement/semantic_repository_enabler.html)
Used by Open Callers, suggestions and comments
Open Caller name: MANTRA **Suggestion and comments: N/A**

4.1.1.3.2. Semantic Translation enabler

Enabler: *Semantic Translation*	**Id:** *T43E2*
Owner and Support: *SRIPAS*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
Semantic Translation enabler offers a configurable service to change the contents of semantically annotated data in accordance with translation rules – so called "alignments". Data can be translated in batch, or through persistent streams.	
Keywords / Key components	
Semantic translation, streaming, ontologies, RDF	
Link for wiki	
Wiki:[*https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/datamanagement/semantic_translation_enabler.html*](https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/datamanagement/semantic_translation_enabler.html)	
Used by Open Callers, suggestions and comments	
Open Caller name: N/A **Suggestion and comments: N/A**	

4.1.1.3.3. Semantic Annotation enabler

Enabler: *Semantic annotation*	**Id:** *T43E3*
Owner and Support: *SRIPAS*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler offers a syntactic transformation service, that annotates data in various formats and lifts it into RDF. Full list of formats is yet to be decided and the first version will support JSON, with CSV and XML to follow Annotation can be done in batch (in first release) or through persistent configurable streams.	
Keywords / Key components	
Semantic annotation, RDF, streams	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/datamanagement/semantic_annotator_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: N/A **Suggestion and comments: N/A**	

4.1.1.3.4. Edge Data Broker enabler

Enabler: *Edge Data Broker*	**Id:** *T43E4*
Owner and Support: *ICCS, UPV*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Testing Phase*	
Enabler Description	
It enables the efficient management of data demand and data supply from/to the Edge Nodes. It optimally distributes data where it is needed for application, services, and further analysis. Data distribution is based on reported demand and available resources at the Edge Nodes. It provides: subscriptions and messages between the broker and the Edge Nodes; management of message scheduling, routing and delivery; common interfaces for searching and finding information.	
Keywords / Key components	
edge data broker, distributed, clustered, MQTT, middleware, data analytics	

Link for wiki
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/datamanagement/edge_data_broker_enabler.html
Used by Open Callers, suggestions and comments
Open Caller name: BREATHE, MANTRA, IOTLORAMESH, Hopcast, CHEaaS **Suggestion and comments: N/A**

4.1.1.3.5. Long-term Data Storage enabler

Enabler: *Long-term Data Storage*	**Id:** *T43E5*
Owner and Support: *PRO, UPV*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
The role of this enabler is to serve as a secure and resilient storage, offering different storage sizes and individual storage space for other enablers (which could request back when they are being initialising in Kubernetes pods). It also guarantees that the data will be kept safe, in face of various kinds of unauthorised access requests, or hardware failures, by only allowing access to the data once the Identity Manager and the Authorisation enablers have confirmed their access rights.	
Keywords / Key components	
Long-Term Storage, noSQL, SQL, resilient, centralized	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/datamanagement/long_term_data_storage_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: BREATHE, HAIR, RAZOR, SMART SONIA, ATHEMS, ADDICTIVE **Suggestion and comments: N/A**	

4.1.1.4. Application and Services Plane

4.1.1.4.1. Tactile Dashboard enabler

Enabler: *Tactile Dashboard*	**Id:** *T44E1*
Owner and Support: *PRO, UPV*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
The Tactile Dashboard enabler has the capability of representing data stored in the ASSIST-IoT pilots, through meaningful combined visualisations in real time. It also provides (aggregates and homogenises) all the User Interfaces for the configuration of the different ASSIST-IoT enablers, and associated components.	
Keywords / Key components	
Frontend, dashboard, VUE.js, responsive, webpage	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/application/tactile_dashboard_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: ATHEMS **Suggestion and comments: N/A**	

4.1.1.4.2. Business KPI Reporting enabler

Enabler: *Business KPI Reporting*	**Id:***T44E2*
Owner and Support: *PRO*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler will illustrate valuable KPIs within Graphical User Interfaces embedded into the tactile dashboard. It will facilitate the visualisation and combination of charts, tables, maps, and other visualisation graphs in order to search for hidden insights.	
Keywords / Key components	
pie charts, bar graphs, KPIs	

Link for wiki
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/application/business_kpi_reporting_enabler.html
Used by Open Callers, suggestions and comments
Open Caller name: ATHEMS **Suggestion and comments: N/A**

4.1.1.4.3. Performance and Usage Diagnosis enabler

Enabler: *Performance and Usage Diagnosis*	**Id:** *T44E3*
Owner and Support: *PRO, UPV*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
Performance and Usage Diagnosis (PUD) enabler aims at collecting performance metrics from monitored targets by scraping metrics HTTP endpoints on them and highlighting potential problems in the ASSIST-IoT platform, so that it could autonomously act in accordance or to notify to the platform administrator to fine tune machine resources.	
Keywords / Key components	
monitoring, metrics collection, targets, status alerting	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/application/performance_and_usage_diagnosis_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: ATHEMS **Suggestion and comments: N/A**	

4.1.1.4.4. OpenAPI Management enabler

Enabler: *OpenAPI Management*	**Id:** *T44E4*
Owner and Support: *CERTH, UPV*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	

Status of Enabler: *Completed*
Enabler Description
The OpenAPI management enabler will be an API Manager that allows enablers that publish their APIs, to monitor the interfaces lifecycles and also make sure that needs of external third parties (including granted open callers), as well as applications that are using the APIs, are being met.
Keywords / Key components
API, open calls, swagger, swagger-json
Link for wiki
Wiki:[*https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/application/openapi_management_enabler.html*](https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/application/openapi_management_enabler.html)
Used by Open Callers, suggestions and comments
Open Caller name: N/A **Suggestion and comments: N/A**

4.1.1.4.5. Video Augmentation enabler

Enabler: *Video Augmentation*	**Id:** *T44E5*
Owner and Support: *PRO, UPV*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler receives data (mainly images or video streams) captured either from ASSIST-IoT Edge nodes, or from ASSIST-IoT databases, and by means of Machine Learning Computer Vision functionalities, it provides object detection/recognition of particular end-user assets (e.g., cargo containers, cars' damages).	
Keywords / Key components	
Object detection, Camera software, AV, ML	
Link for wiki	
Wiki:[*https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/application/video_augmentation_enabler.html*](https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/application/video_augmentation_enabler.html)	
Used by Open Callers, suggestions and comments	
Open Caller name: SPINE **Suggestion and comments: More frequent updates on GitLab Repository**	

4.1.1.4.6. MR enabler

Enabler: *MR*	**Id:** *T44E6*
Owner and Support: *ICCS*	
Related Deliverable/s: *D4.1, D4.2, D4.3*	
Status of Enabler: *Completed*	
Enabler Description	
The novel interface that is used in the MR enabler offers a human-centric interaction through better cooperation of the end-users with the IoT environment. Through the MR enabler, the human effort and decisions are introduced in the loop of every critical action, whenever needed. The MR enabler aids human-friendly haptics and the end-user can receive and provide tactile, real-time and visual feedback as well as data capable of identifying critical improvements, preventions and triggers in long-, short-term, or real-time. Through reporting functions, the MR enabler gathers reliable data to extract information and perform analytics. Decision-making is improved as human flexibility, creativity and expertise, interact with IoT platforms and devices. The functionalities of the MR enabler are summarized as follow: • *Identifying assets (along with relevant data) at close proximity,* • *Visualizing rendered (3D) models through the head-mounted MR devices, along with highlighted zones of the same model. The models and all related data come from the long-term storage,* • *Shows location and information of the workers of the construction site,* • *Receiving alert messages from real-time data streams and displaying them to the user, and* • *Capturing and storing media files in order to include them in a report.*	
Keywords / Key components	
Mixed reality, BIM visualisation, real-time data, IoT, alerting	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/horizontal_planes/application/mr_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: N/A **Suggestion and comments: N/A**	

4.1.2. Verticals' Enablers

4.1.2.1. Self-* Enablers

4.1.2.1.1. Self-healing device enabler

Enabler: *Self-healing device*	**Id:***SELF11*
Owner and Support: *PRO, UPV*	

Related Deliverable/s: *D5.1, D5.3, D5.4, D5.5*
Status of Enabler: *Completed*
Enabler Description
This enabler aims at providing to IoT devices with the capabilities of actively attempting to recover themselves from abnormal states, mainly divided in three categories: 1) security (jamming, DoS), 2) dependability (data corruption, network protocol violation), and 3) long-term (HW's end-of-life, HW unsupported capabilities), based on a pre-established routine schedule.
Keywords / Key components
IDS, RAM monitoring, CPU monitoring. self-healing
Link for wiki
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/self/self_healing_device_enabler.html
Used by Open Callers, suggestions and comments
Open Caller name: N/A **Suggestion and comments: N/A**

4.1.2.1.2. Resource provisioning enabler

Enabler: *Resource Provisioning*	**Id:***SELF12*
Owner and Support: *UPV*	
Related Deliverable/s: *D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler will be able to horizontally scale (up or down) the resources devoted to a specific enabler (inside a node) in a dynamic fashion, based on time series inference and custom logic.	
Keywords / Key components	
Self-configuration, Time Series, Horizontal Pod Autoscaler	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/self/resource_provisioning_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: N/A	

Suggestion and comments: N/A

4.1.2.1.3. Location processing enabler

Enabler: *Location Processing*	**Id:***SELF16*
Owner and Support: *SRIPAS*	
Related Deliverable/s: *D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler provides spatial data storage and processing capabilities. It is able to integrate spatial information from various sources and process it in real time, in a streaming fashion.	
Keywords / Key components	
Location, Database, Query, Data Streaming	
Link for wiki	
Wiki:*https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/self/location_process_enabler.html*	
Used by Open Callers, suggestions and comments	
Open Caller name: N/A **Suggestion and comments: N/A**	

4.1.2.1.4. Monitoring and Notifying enabler

Enabler: *Monitoring and Notifying*	**Id:***SELF14*
Owner and Support: *CERTH, SRIPAS*	
Related Deliverable/s: *D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler could be viewed as a general purpose by representing it as a combination of high-level monitoring module (which would allow to monitor devices, logs, etc.) and notifying a module that could send custom messages to predefined system components.	
Keywords / Key components	
Monitoring, Notifying, Data Streaming, Message Queue	

Link for wiki
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/self/monitoring_and_notifying_enabler.html
Used by Open Callers, suggestions and comments
Open Caller name: ATHEMS, Hopcast, CHEaaS **Suggestion and comments: N/A**

4.1.2.1.5. Automated configuration enabler

Enabler: *Automated configuration*	**Id:***SELF15*
Owner and Support: *SRIPAS*	
Related Deliverable/s: *D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
Automated Configuration Enabler keeps heterogenous devices and services synchronised with their configurations. User can update configuration and define fallback configurations in case of errors. Self- component will be responsible for reacting to changing environment and updating configuration as necessary.*	
Keywords / Key components	
Configuration, Self-Management, Synchronization	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/self/automated_configuration_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: ATHEMS **Suggestion and comments: N/A**	

4.1.2.2. Federated machine learning enablers

4.1.2.2.1. FL Orchestrator

Enabler: *FL Orchestrator*	**Id:***T52E1*
Owner and Support: *PRO, SRIPAS*	
Related Deliverable/s: *D5.1, D5.2, D5.3, D5.4, D5.5*	

Status of Enabler: *Completed*
Enabler Description
The FL orchestrator is responsible of specifying details of FL workflow(s)/pipeline(s). This includes FL job scheduling, managing the FL life cycle, selecting, and delivering initial version(s) of the shared algorithm, as well as modules used in various stages of the process, such as training stopping criteria. Finally, it can specify ways of handling different "error conditions" that may occur during the FL process.
Keywords / Key components
Orchestrator, Federated Learning, Lifecycle.
Link for wiki
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/federated/fl_orchestrator.html
Used by Open Callers, suggestions and comments
Open Caller name: HazardMiner, Poseidon **Suggestion and comments: N/A**

4.1.2.2.2. FL Training Collector

Enabler: *FL Training Collector*	**Id:** *T52E2*
Owner and Support: *SRIPAS, PRO*	
Related Deliverable/s: *D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
The FL training process involves several independent parties that commonly collaborate in order to provide an enhanced ML model. In this process, the different local update suggestions shall be aggregated accordingly. This duty within ASSIST-IoT will be tackled by the FL Training Collector, which will also be in charge of delivering back the updated model. The FL training collector will consist of two components: (i) the combiner responsible of providing updates with respect to the shared averaged model, and (ii) the I/O component which will carry out the input and output communications of the enabler.	
Keywords / Key components	
Federated Learning, Model Update, Aggregator, Model Enhancement	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/federated/fl_training_collector.html	

Used by Open Callers, suggestions and comments
Open Caller name: HazardMiner, Poseidon
Suggestion and comments: N/A

4.1.2.2.3. FL Repository

Enabler: *FL Repository*	**Id:** *T52E3*
Owner and Support: *SRIPAS, PRO*	
Related Deliverable/s: *D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
The FL repository will be a set of different databases, including initial ML algorithms, already trained ML models suitable for specific data sets and formats, averaging approaches, and auxiliary repositories for other additional functionalities that may be needed, and are not specifically identified yet.	
Keywords / Key components	
Federated Learning, Repository, Model Storage	
Link for wiki	
Wiki: https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/federated/fl_repository.html	
Used by Open Callers, suggestions and comments	
Open Caller name: HazardMiner, Poseidon	
Suggestion and comments: N/A	

4.1.2.2.4. FL Local Operations

Enabler: *FL Local Operations*	**Id:** *T52E4*
Owner and Support: *SRIPAS, PRO*	
Related Deliverable/s: *D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
FL Local Operations enabler is an embedded enabler within each FL involved party/device of the FL systems. The FL Local Operation enabler will consist of four components: Local Data Transformer component (that will be in charge of guaranteeing that data is appropriately formatted for the FL model in use), Local Model	

Training component, Local Model Inference component, and Communication component (to enable in and out communications between involved local parties and FL orchestrator and FL collector).
Keywords / Key components
Federated Learning, Local Training, Local Inferencing, FL Party
Link for wiki
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/federated/fl_local_operations.html
Used by Open Callers, suggestions and comments
Open Caller name: HazardMiner, Poseidon **Suggestion and comments: N/A**

4.1.2.3. Cybersecurity enablers

4.1.2.3.1. Authorization enabler

Enabler: *Authorisation*	**Id:***T53E1*
Owner and Support: *S21SEC*	
Related Deliverable/s: *D4.1, D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
Authorisation server offers a decision-making service based on XACML policies. It has different modules that interact and can be deployed independently such as, PEP (Policy Enforcement Point), PAP (Policy Administration Point), PIP (Policy Information Point) and PDP (Policy Decision Point).	
Keywords / Key components	
XACML, PEP, PAP, PIP, PDP, policies	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/cybersecurity/authorization_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: N/A **Suggestion and comments: N/A**	

4.1.2.3.2. Identity Manager enabler

Enabler: *Identity Manager*	**Id:***T53E2*
Owner and Support: *S21SEC*	
Related Deliverable/s: *D4.1, D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
Using OAuth2 protocol, it will offer a federated identification service where service requester and provider will be able to establish a trusted relation without previously knowing each other. This way a secure identification process is completed without the service provider having received the requester credentials.	
Keywords / Key components	
OAuth2, federated, trusted, secure, credentials	
Link for wiki	
Wiki:[*https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/cybersecurity/identity_manager_enabler.html*](https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/cybersecurity/identity_manager_enabler.html)	
Used by Open Callers, suggestions and comments	
Open Caller name: Hopcast, CHEaaS **Suggestion and comments: N/A**	

4.1.2.3.3. Cybersecurity Monitoring enabler

Enabler: *Cybersecurity Monitoring*	**Id:***T53E3*
Owner and Support: *S21SEC*	
Related Deliverable/s: *D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
Cybersecurity monitoring enabler, provides security awareness, visibility and infrastructure monitoring. Having raw data as input, the enabler will set a series of processing steps that will enable the discovery of cybersecurity threats, going through a sequence step: (i) collecting, parsing, and normalising input events, (ii) enriching normalised events, (iii) correlating events for detecting cybersecurity threats.	
Keywords / Key components	
Security, agentless, monitoring, discovery, threats, normalising, cybersecurity, detecting	

Link for wiki
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/cybersecurity/cybersecurity_monitoring_enabler.html
Used by Open Callers, suggestions and comments
Open Caller name: N/A **Suggestion and comments: N/A**

4.1.2.3.4. Cybersecurity Monitoring Agent enabler

Enabler: *Cybersecurity Monitoring Agent*	**Id:***T53E4*
Owner and Support: *S21SEC*	
Related Deliverable/s: *D4.1, D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
Perform functions of an endpoint detection and response system, monitoring and collecting activity from end points that could indicate a threat. Security agent runs at a host-level, combining anomaly and signature-based technologies to detect intrusions or software misuse.	
Keywords / Key components	
Endpoint, response, detection, collecting, host-level, anomaly, intrusions, monitoring	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/cybersecurity/cybersecurity_monitoring_agent_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: N/A **Suggestion and comments: N/A**	

4.1.2.4. DLT-based enablers

4.1.2.4.1. Logging and auditing enabler

Enabler: *Logging and Auditing*	**Id:***T54E1*
Owner and Support: *CERTH*	
Related Deliverable/s: *D5.1, D5.2, D5.3, D5.4, D5.5*	

Status of Enabler: *Completed*
Enabler Description
This enabler will log critical actions that happen during the data exchange between ASSIST-IoT stakeholders to allow for transparency, auditing, non-repudiation and accountability of actions during the data exchange. It will also log resource requests and identified security events to help to provide digital evidence and resolve conflicts between stakeholders, when applicable.
Keywords / Key components
Logging, Auditing, DLT-based
Link for wiki
Wiki:[*https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/dlt/logging_and_auditing_enabler.html*](https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/dlt/logging_and_auditing_enabler.html)
Used by Open Callers, suggestions and comments
Open Caller name: N/A **Suggestion and comments: N/A**

4.1.2.4.2. Data integrity verification enabler

Enabler: *Data Integrity Verification*	**Id:***T54E2*
Owner and Support: *CERTH*	
Related Deliverable/s: *D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
This is an enabler responsible for providing DLT-based data integrity verification mechanisms that allow data consumers to verify the integrity of any data at question. Network peers host smart contract (chaincode) which includes the data integrity business logic. It stores hashed data in a data structure and it compares it with the hashed data of the queries made by clients in order to verify their integrity.	
Keywords / Key components	
Verification, DLT-based	
Link for wiki	
Wiki:[*https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/dlt/data_integrity_verification_enabler.html*](https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/dlt/data_integrity_verification_enabler.html)	
Used by Open Callers, suggestions and comments	
Open Caller name: N/A	

Suggestion and comments: N/A

4.1.2.4.3. Distributed broker enabler

Enabler: *Distributed Broker*	**Id:** *T54E3*
Owner and Support: *CERTH*	
Related Deliverable/s: *D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler will provide a mechanism that will facilitate data sharing between heterogeneous IoT devices belonging to various edge domains and/or between different enablers of the architecture. Using the security features of DLT, it will deal with data source metadata management and provide to the data consumers trustable, findable and retrievable metadata for the data sources.	
Keywords / Key components	
Sharing, DLT-based, metadata, Management	
Link for wiki	
Wiki: https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/dlt/distributed_broker_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: Hopcast **Suggestion and comments: N/A**	

4.1.2.4.4. DLT-based FL enabler

Enabler: *DLT-based FL*	**Id:** *T54E4*
Owner and Support: *CERTH*	
Related Deliverable/s: *D5.1, D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler is a system that uses blockchain technology to provide a secure reputation mechanism for local operators in a federated learning (FL) system. The reputation mechanism prevents free-riders and malicious adversaries from accessing the global model without contributing to it. The system calculates reputation scores for each local operator instance and stores them on a permissioned blockchain network, ensuring privacy and security. The FL training collector uses the scores to decide on penalties or incentives for the	

FL local operations. The FL-DLT enabler consists of three components: the DLT communicator, the reputation score calculator, and the DLT storage.
Keywords / Key components
FL, DLT-based, Decentralised, Models, Validation, Exchange
Link for wiki
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/dlt/dlt_based_fl_enabler.html
Used by Open Callers, suggestions and comments
Open Caller name: N/A **Suggestion and comments: N/A**

4.1.2.5. Manageability

4.1.2.5.1. Enablers manager

Enabler: *Enablers manager*	**Id:***T55E1*
Owner and Support: *UPV*	
Related Deliverable/s: *D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler will serve as a registry of enablers and, in case they are deployed, the retrieval of their status. In particular, it will: (a) Allow the registration of an enabler (this is, from an ASSIST-IoT repository). Essential enablers will be pre-registered, (b) Retrieve a list of currently-running enablers, (c) Depict the status and the specific logs of an enabler (the latter only if the enabler with log collection capabilities is in place), (d) facilitate the deployment of standalone enablers (mostly for those that have to be present at any deployment).	
Keywords / Key components	
Enablers registration, Enablers status, Helm repository, GUI	
Link for wiki	
Wiki:https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/manageability/registration_and_status_enabler.html	
Used by Open Callers, suggestions and comments	
Open Caller name: ATHEMS **Suggestion and comments: N/A**	

4.1.2.5.2. Composite services manager

Enabler: *Management of services and enablers workflow*	**Id:** *T55E2*
Owner and Support: *UPV*	
Related Deliverable/s: *D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
This enabler will present a graphical environment where ASSIST-IoT administrators can instantiate the enablers required to work, and also to connect them to compose a composite service (i.e., a workflow). Having information about the physical topology and available k8s nodes/clusters, it will allow the user to decide whether to select the proper node or cluster for deploying an enabler, or let the system decide based on pre-defined architectural rules.	
Keywords / Key components	
Service composition, GUI	
Link for wiki	
Wiki:[*https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/manageability/registration_and_status_enabler.html*](https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/manageability/registration_and_status_enabler.html)	
Used by Open Callers, suggestions and comments	
Open Caller name: N/A **Suggestion and comments: N/A**	

4.1.2.5.3. Clusters and topology manager

Enabler: *Clusters and topology manager*	**Id:** *T55E3*
Owner and Support: *UPV*	
Related Deliverable/s: *D5.2, D5.3, D5.4, D5.5*	
Status of Enabler: *Completed*	
Enabler Description	
Integrated in the tactile dashboard, the main functionality of this enabler is to register new clusters of computing nodes (or a single computing node) in an ASISST-IoT deployment, and present an overview of its topology. These nodes require to have a functional and accessible K8s deployment.	
Keywords / Key components	

K8s, Cluster registration, IoT device registration, GUI
Link for wiki
Wiki:[*https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/manageability/devices_management_enabler.html*](https://assist-iot-enablers-documentation.readthedocs.io/en/latest/verticals/manageability/devices_management_enabler.html)
Used by Open Callers, suggestions and comments
Open Caller name: ATHEMS **Suggestion and comments: N/A**

4.2. Open Callers (Round 2) Feedback Compilation

To gain a comprehensive understanding of how Open Callers #2 perceive the ASSIST-IoT Technical and Support Documentation as well as the GitLab Repository, we designed again an online questionnaire for Open Call 2 participants this time. This survey played a pivotal role in improving the documentation's user-friendliness and functionality by offering valuable insights. It also helped us to further pinpoint any areas of confusion or challenges users encountered, enabling us to address these issues, enhance the Documentation and GitLab Repository, and incorporate requested features. In summary, the questionnaire proved to be an invaluable tool for enhancing the enablers' effectiveness and delivering top-notch documentation to users.

In a separate section of the survey, Open Callers were asked for their overall impressions of the GitLab Repository. Based on the feedback received, our objective is to assess the ease of accessing and managing artifacts, identify areas in need of improvement, and gather recommendations for optimization. This information will be used to further refine the GitLab Repository, ensuring that it seamlessly integrates into users' workflows.

This survey was created using Microsoft Forms, an online survey and quiz development tool by Microsoft Corporation. This software is designed to simplify the creation of forms, surveys, quizzes, and polls for targeted audiences. Microsoft Forms offers a wide range of question types, making it quick to develop surveys and quizzes while also allowing for easy collaboration with others. Additionally, this tool provides robust analytics and reporting capabilities, enabling us to extract meaningful insights from the collected data.

4.2.1. Context and Target Audience of the Questionnaire

Questionnaire participants: the eight winners of Open Call 2

The table below presents in detail the questions used and the answer type to each question.

Table 58: Questionnaire Context

Question	Answer Type
ASSIST-IoT Documentation Section:	
1) OC-Project title:	Free Text
2) Have you used or are you planning to use the technical documentation provided here: *https://assist-iot-enablers-documentation.readthedocs.io/en/latest/*	Yes/No
3) Does the documentation provide you with enough information to understand and utilize the enablers?	Rating Scale 1-5
4) Do you have any suggestions to improve or update the documentation?	Free Text
5) Do you find the documentation user-friendly?	Rating Scale 1-5
6) Which enablers do you plan to use?	Multiple Choice

7) How good did you find the documentation in our deliverables when preparing your proposal?	Free Text
Technical Scope of ASSIST-IoT Section:	
8) How different did you feel about the technical scope of ASSIST-IoT enablers and architecture comparing to now and when you were writing your proposal?	Free Text
9) How do you manage to fulfil any unexpected gap between the promised ASSIST-IoT technical scope in comparison what you have now available?	Free Text
GitLab Section:	
10) Did you find the GitLab Repository useful?	Rating Scale 1-5
11) Do you have any prior experience using and managing GitLab artifacts?	Rating Scale 1-5
12) How would you rate the accessibility process at GitLab repositories?	Rating Scale 1-5
13) Based on your general experience using GitLab, would you have any recommendations to suggest?	Free Text

4.2.2. Questionnaire Results

The purpose of this section is to share the results of the questionnaire, emphasize how the comments were handled, and showcase any enhancements that were made. Additionally, you can find a comprehensive presentation of the responses received by Open Call 2 participants (in total 8 participants) in **A: Open Callers' Questionnaire Responses.**

The chart Enablers used by Open Callers shows which enablers they favoured the most. This statistical breakdown allows us to pinpoint the most popular enablers. Moreover, the feedback gathered from the questionnaire is rooted in the experiences of Open Callers with these enablers. Additionally, it is worth noticing that enablers selected in the second round of Open Calls, differ from those selected in the first round.

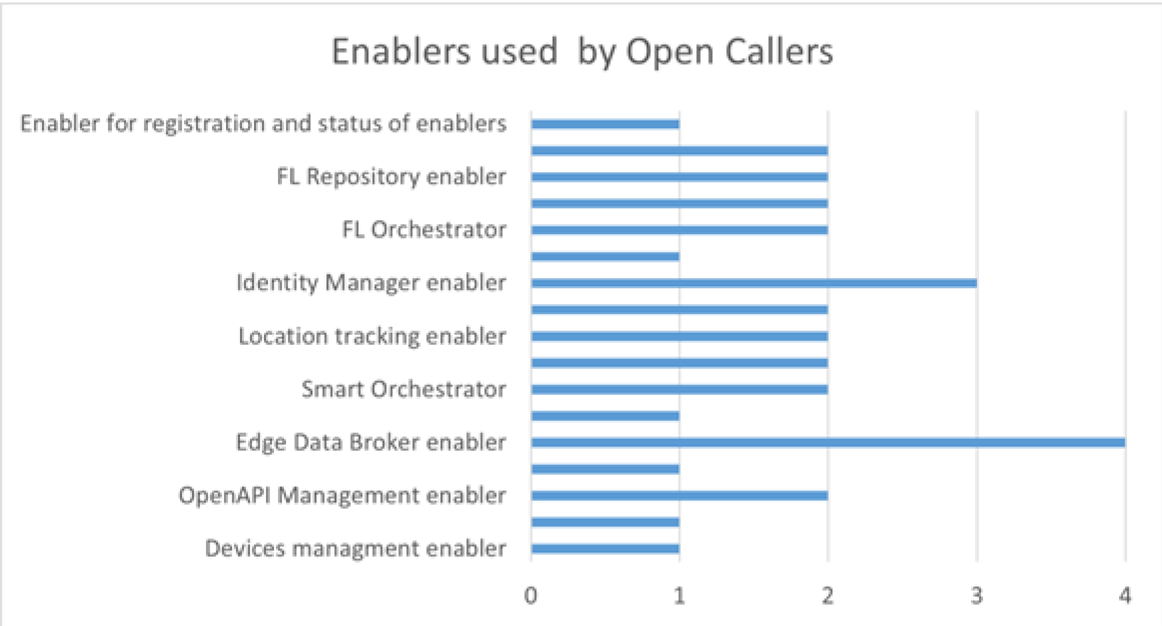


Figure 6. Enablers used by Open Callers 2

4.2.2.1. ASSIST-IoT Documentation Section Feedback

To continually enhance and refine our Technical and Support Documentation, it is crucial to gauge the satisfaction of our users. As such, we have gathered and analysed data concerning the satisfaction and feedback from Open Callers #2 to gain insights into how well our documentation meets their requirements. In this section, we will present the outcomes of our analysis, encompassing user satisfaction statistics and direct feedback from Open Callers #2. This information has aided us in identifying areas for improvement and in elevating the overall quality and user-friendliness of our documentation.

The pie chart below (**Technical Documentation Usage**) visually represents the extent to which Open Callers have utilized or intend to use the ASSIST-IoT Technical and Support Documentation. It reveals that seven out of eight Open Callers either used the documentation or plan to do so, indicating a high level of engagement with the Enablers. This suggests that the documentation was clear and provided users with the necessary information to effectively employ the Enablers. The one Open Caller who did not utilize the documentation will manage to learn about the Enablers through interactions and communication with the Pilot Host.

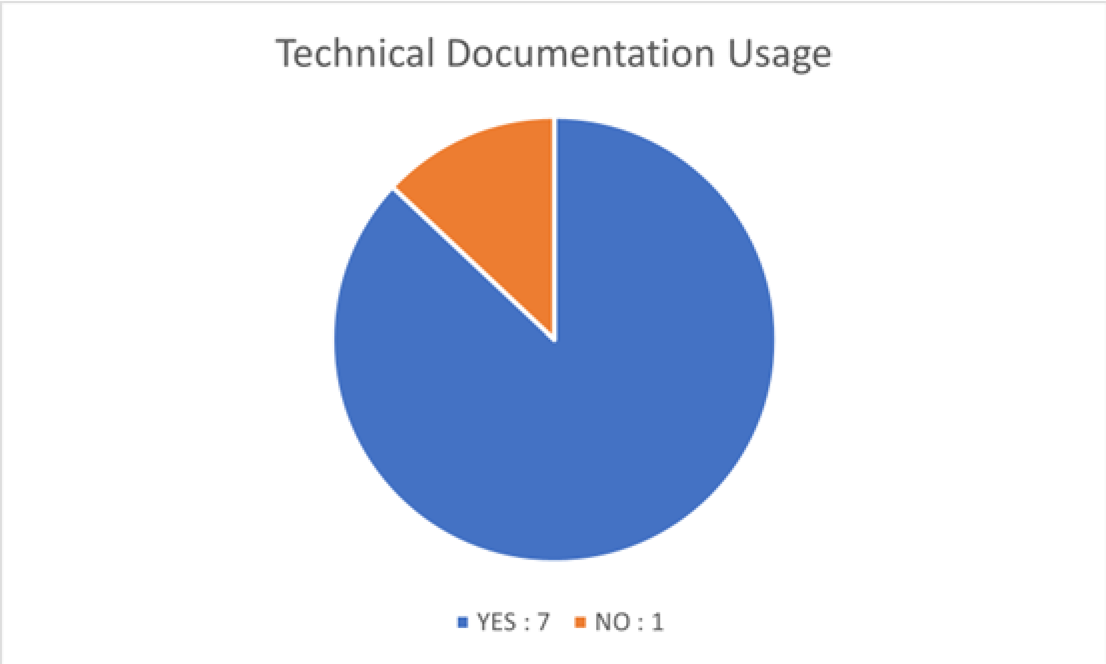


Figure 7. Technical Documentation Usage

The chart titled **Understanding and Utilizing Enablers** illustrates the feedback received from individuals who have participated in Open Call 2. It reflects their capacity to grasp and put into practice the enablers as per the documentation provided. The primary objective of this chart is to demonstrate the documentation's efficacy in aiding Open Call #2 participants in comprehending and implementing the enablers. The rating of 3.85 out of 5 for this question signifies that the documentation was perceived as beneficial by Open Call participants for understanding and using the enablers. This positive assessment suggests that the documentation is both thorough and valuable.

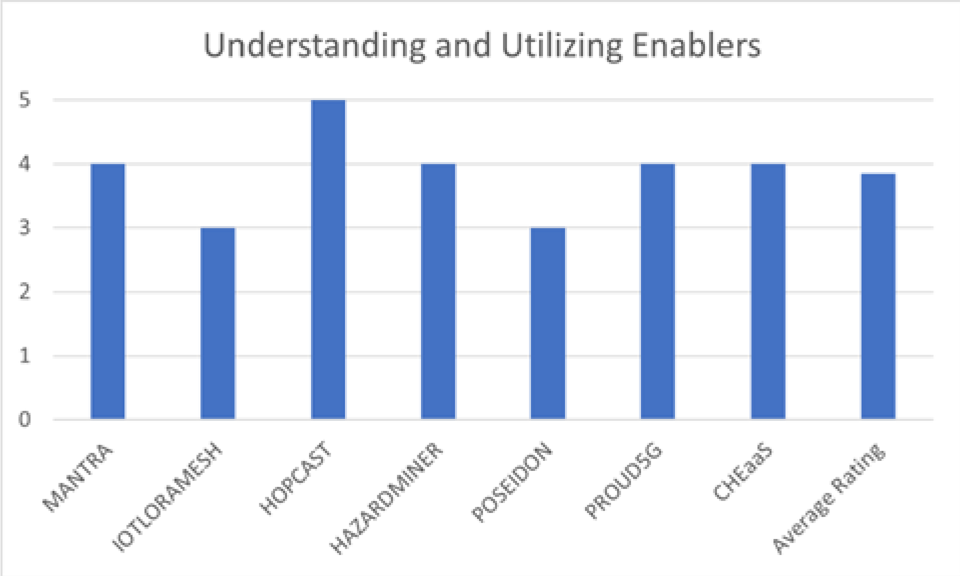


Figure 8. Understanding and Utilizing Enablers

According to the feedback gathered from the questionnaire (**Documentation User-Friendliness**), specifically regarding the user-friendliness of ASSIST-IoT Technical and Support Documentation, we are delighted to announce that it has received high ratings for its ease of use. On average, it has been rated at 4 out of 5, signifying that most Open Callers2 find the documentation straightforward to navigate. This rating underscores the importance of user-friendliness in successful documentation and highlights the effectiveness of the documentation's design.

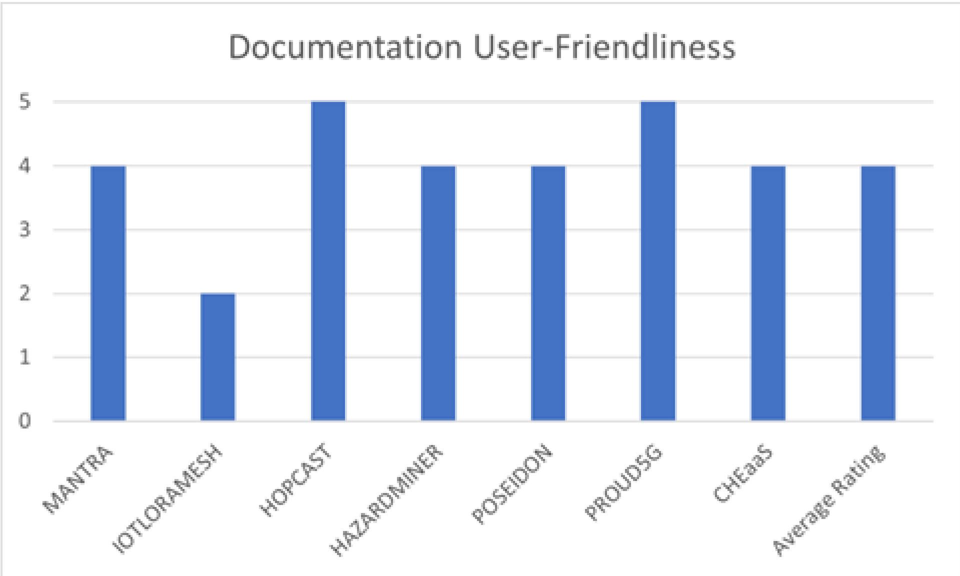


Figure 9. Documentation User-Friendliness

4.2.2.1.1. Suggestions from Open Callers #2 and Improvements made by ASSIST-IoT team

The following section summarizes the suggestions and enhancements that have been put into practice based on feedback from Open Callers. This section encompasses a variety of ideas aimed at improving the user experience, along with the actions being taken to implement them.

It is worth mentioning that the comments received in this round of feedback differ from those in the first round of Open Call, signifying that the suggestions made in the first round have been addressed accordingly and some new aspects needed to be considered and resolved. Regarding the survey question, users shared valuable insights on how to further improve the documentation.

1. Enhance Component Deployment Information:
    - Provide additional information about deploying the application components.
2. Improve Federated Learning Enabler Documentation:
    - Enhance documentation related to Federated Learning enablers.
    - Include detailed configurations required for different services to be connected.
    - Provide more information on API payloads.
    - Include procedures for testing the proper installation of the services.
3. Enhance Multi-link Enabler Description:
    - Offer a more comprehensive description of the Multi-link Enabler.
4. Acknowledgment of Well-Structured Documentation:
    - Users appreciated the overall structure of the documentation.
    - Mentioned that some sections need to be completed but found it to be well-organized.

The feedback related to documentation has been taken into account, and efforts are underway to address those concerns before Open Call 2 and project's end. As for the comments concerning application-related topics, particularly the tutorial guides, they have been shared with the Pilot leaders for their consideration and action.

4.2.2.2. Technical Scope-Section of ASSIST-IoT Feedback

This section of the questionnaire aimed to collect feedback from Open Callers 2 regarding their views on the technical aspects of the ASSIST-IoT enablers and architecture, and how these views evolved during the project. Specifically, participants were asked to compare their initial understanding of the project's technical scope with their perception after working on the project. Additionally, Open Callers were inquired about how they addressed any unanticipated disparities between the promised technical scope and their actual experiences.

The responses to these questions will be utilized to better grasp how effectively the technical scope of the ASSIST-IoT project was communicated and comprehended by Open Callers. It also serves the purpose of pinpointing areas for potential improvement.

By examining the answers to these questions, we can gain insights into the factors that influenced participants' perceptions of the project's technical scope, as well as how adeptly they handled unexpected challenges. This information can be harnessed to enhance the project, ensuring that Open Callers possess a comprehensive understanding of the technical scope and are well-prepared to tackle any unforeseen challenges that may arise.

4.2.2.2.1. Open Callers' Perception of Technical Scope and Enabler Architecture

In analysing the responses from Open Callers #2 who were queried about their perceptions of the technical scope of the ASSIST-IoT project, it becomes evident that their experiences are diverse and offer valuable insights. The feedback ranges from a significantly clearer understanding of the project's technical aspects to expectations and misunderstandings that have arisen during their interaction with the project. Users highlight the importance of effective documentation, transparent communication, and continual improvement of the project's architecture and enablers. While some express positive views of project evolution, others stress the need for better alignment between user expectations and the project's capabilities. These responses underscore the importance of maintaining clear, comprehensive, and accessible documentation, along with an emphasis on addressing user expectations and misunderstandings, in order to enhance the overall user experience within the ASSIST-IoT project.

- **MOTION:** Feels they now have a much clearer understanding of the technical scope of ASSIST-IoT. This indicates positive progress in their comprehension of the project.

- **MANTRA:** Believes the technical scope is the same as when they wrote the proposal. This suggests a perception of consistency or perhaps a need for further communication.
- **IOTLORAMESH:** Recognizes the completeness of the architecture but notes that enabler implementation depends on hardware capacities. This highlights a potential limitation in the project.
- **HOPCAST:** Finds it difficult to make a direct comparison due to the time gap but appreciates the helpful documentation. Documentation seems to have a positive impact on their understanding.
- **HAZARDMINER:** Indicates that the technical scope has evolved and improved since writing the proposal, demonstrating positive development in the project.
- **POSEIDON:** Expresses an expectation that certain enablers (FL enablers) would be ready but notes a delay. This could be a concern, especially if the project is nearing completion.
- **PROUD5G:** Acknowledges a misunderstanding of the functionality of specific enablers (Multi-Link enabler) during the proposal writing, leading to necessary corrections. Clearer communication may be needed.
- **CHEaaS:** Feels their understanding of ASSIST-IoT has remained nearly the same before and after the proposal. Interaction with project participants has provided a better understanding, but there is a desire for further exploration and concrete use cases. This suggests room for deeper engagement and more specific application scenarios.

In summary, the Open Callers' responses provide valuable insights into their experiences with the ASSIST-IoT project. These insights can guide ongoing efforts to enhance communication, and user engagement, ultimately improving the user experience and fostering a more profound understanding of the project's technical scope.

4.2.2.2.2. Managing Unexpected Gaps in ASSIST-IoT Technical Scope, Open Callers Experiences – Suggestions and Improvements

It is evident that managing unexpected gaps between the promised ASSIST-IoT technical scope and the available resources can be a complex task. To address this challenge, the responses provided by the Open Callers demonstrate a proactive approach aimed at meeting the project's goals. These responses highlight a range of strategies, including technical adaptation, communication with ASSIST-IoT team, and innovative workarounds.

The responses provided by the Open Callers, offer a comprehensive view of their approach to managing unexpected gaps in the ASSIST-IoT project. They reveal a commitment to meeting the technical scope, even when faced with unique challenges. Some are finding creative solutions, like emulating vehicle-to-vehicle communications with Android devices, while others focus on enhancing documentation and theoretical understanding. This diversity of strategies demonstrates adaptability and a proactive attitude.

In conclusion, the responses from the Open Callers exemplify a dynamic and proactive approach to managing unexpected gaps in the ASSIST-IoT project. Their unwavering commitment to meeting the technical scope, coupled with creative problem-solving, fosters an environment of innovation. Furthermore, the strong emphasis on clear communication and collaboration with the ASSIST-IoT team demonstrates a robust dedication to the project's success. These responses collectively underscore the adaptability and resilience of the teams involved, assuring that any unanticipated gaps are met with effective solutions and a determination to bridge them, ultimately paving the way for the project's continued progress and achievement of its objectives.

4.2.2.3. GitLab Repository Segment in the Questionnaire

Based on feedback from Open Callers who utilized the project's GitLab Repository, it appears that the repository offered a moderate level of utility to them, receiving an average rating of 3.75 out of 5. While this rating may not be exceptionally high, it does suggest that the repository did provide valuable support to the Open Callers.

The average rating of 3.75 out of 5 for their experience in using and managing GitLab artifacts also indicates a moderate level of satisfaction. This implies that while users were able to effectively use the repository, they might have encountered some challenges or difficulties along the way.

On a more positive note, the high average rating of 4 out of 5 for the accessibility of the GitLab repository process suggests that it was easy to access and navigate. This indicates that users could quickly and efficiently locate the information they needed, which is a crucial aspect of user satisfaction. You can refer to the **GitLab**

Repository Ratings chart for an overview of the ratings provided by Open Callers who used the project's GitLab Repository.

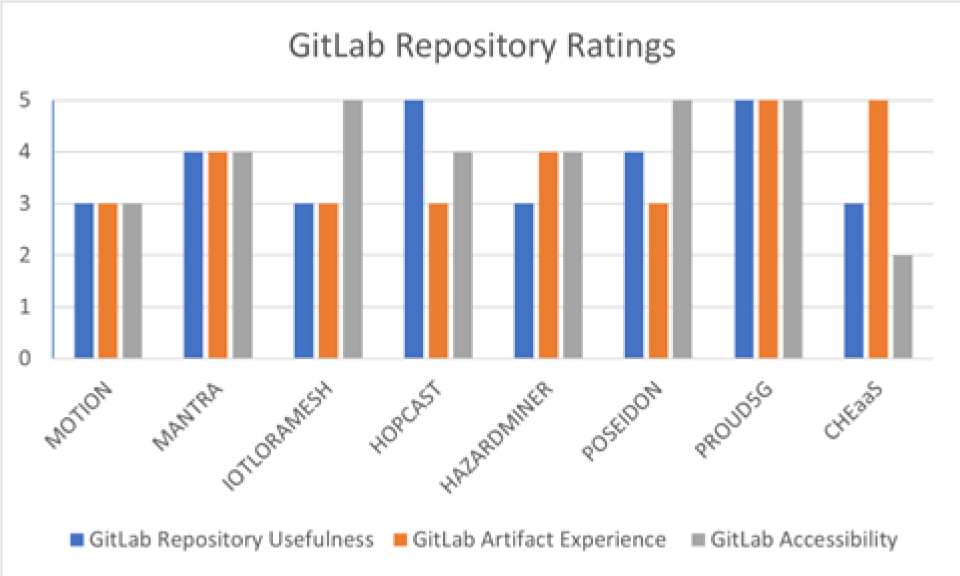


Figure 10. GitLab Repository Ratings

4.2.2.3.1. Suggestions and Improvements

Incorporating Open Callers' feedback is crucial for enhancing any platform, and GitLab is no exception. Open Callers have shared their valuable experiences and thoughts on using GitLab, which can serve as a guide for making the platform even better. In this section, we will address the recommendations provided by Open Callers.

1. **Higher Availability of Source Code**: Some Open Callers expressed interest in leveraging GitLab for higher availability of source code, but noted that the license policies of enabler providers could impact this. While we respect the complexities of licensing, ASSIST-IoT team encourage discussions with these providers to explore possibilities for increasing the availability of source code on GitLab. Collaborative solutions may lead to more open access for Open Callers.
2. **Bugs and Code Testing**: Open Callers who faced challenges with unexpected bugs suggested that deeper code testing before submitting changes to GitLab could help prevent such issues. ASSIST-IoT team, encourage contributors to perform thorough testing and quality assurance before committing their code.
3. **Access Rights Clarity**: Open Callers reported issues with unclear access rights, stating they only had access to public repositories. ASSIST-IoT team acknowledge the importance of clear access permissions and are committed to improving the transparency of access rights.

Overall, these suggestions and improvements could help to enhance the user experience and make it easier for users to understand the state of the project and use GitLab more effectively. By addressing these concerns, GitLab repository can continue to provide a valuable service to its users and improve its overall functionality and usability.

4.2.2.4. Documentation Relevancy for Open Callers' proposals

In assessing the Documentation Relevancy for Open Callers' proposals, it is imperative to consider the feedback provided by Open Callers. Their insights reflect a diverse spectrum of perspectives on the documentation. While some Open Callers found the documentation exceptionally useful, praising its clarity and comprehensiveness, others identified certain areas that required improvement. Notably, the feedback emphasizes the critical role that documentation plays in shaping the quality of proposals submitted.

To sum it up, the feedback suggests that the provided documentation effectively prepared users with the necessary information to create their proposals. You can find a detailed breakdown of the responses from all the Open Call participants in Table 59:

Table 59: Responses on the effectiveness of documentation in proposal developments

Question: How good did you find the documentation in our deliverables when preparing your proposal?	
MOTION	The description of the data and the scanner setup for the Pilot3B is very good, with the exception of a few issues. Due to the large image size and the substantial number of images, it is not feasible to provide a solution that will run efficiently on a Jetson. To address this, we propose to provide a containerized OpenAPI enabler designed to run on Pilot3B's edge device which is very much the same and probably fits better to Pilot3B as it removes the communication overhead between the jetson and the edge node.
MANTRA	It was OK
IOTLORAMESH	The deliverables about the reference architecture were useful.
HOPCAST	Most of the information required was made available through the documentation. Discussions with partners helped clarify few things.
HAZARDMINER	I found the documentation in your deliverables to be clear, comprehensive, and well-organized.
POSEIDON	I found it satisfactory.
PROUD5G	We think deliverables were moderately useful to us. They did not contain all the aspects necessary for us to correctly interpret architecture.
CHEaaS	They were good quality deliverables. However, there were quite high number of information. The online documentation, everything in one place is a much better option.

4.2.2.5. Conclusion to Open Callers' Feedback

In this current round of open calls, we deeply appreciate the invaluable feedback we received from the first group of Open Callers during 1st quarter of 2023. This initial feedback played a significant role in enhancing both our Technical and Support Documentation and the GitLab Repository.

It is noteworthy that during the Open Call 2 and the current questionnaire, we did not encounter the same suggestions or comments for improvements, indicating the positive impact of our previous revisions. Furthermore, it is evident that ASSIST-IoT has diligently and successfully incorporated these suggestions and improvements, as evidenced by the results of of the Open Call 2 questionnaire. The suggestions put forward by Open Callers were effectively addressed in our recent GitLab repository management. Firstly, a link to the external Technical and Support Documentation was included, as suggested. This link has now been incorporated to provide easy access to the relevant documentation, enhancing the comprehensiveness of the repository. Secondly, in response to the desire for more frequent updates, it was ensured that the repository was regularly updated. It has now reached its final version, with all intended features and functionalities fully implemented, offering a clear picture of the project's current state. Lastly, the missing sections in the Technical Documentation were meticulously completed, ensuring that all critical information is now available, improving the overall quality and usefulness of the repository.

Following the same course of actions, we have already started addressing the new suggestions received and implementing updates/changes that will further improve the ASSIST-IoT documentation and repositories.

5. Conclusions / Future Work

This final report of WP6 represents the culmination of efforts within this work package. It provides a comprehensive update on the methodologies and actions outlined in the previous iterations of the three deliverable series. The report provides insight in conceptualization, implementation and outcomes for testing, integration, and support. It builds upon and refines the concepts defined in seven previous deliverables (D6.1-D6.7). While the majority of testing is now complete, for the sake of thorough consolidation, the final wrap-up will also be concluded in D8.3. The repository structure has been finalized, and software licensing and IPR compliance have been documented. Documentation is complete, with refinements to be made based on questionnaire feedback. Given that testing, integration, and support are continuous tasks, ongoing improvements are anticipated. These improvements aim to facilitate both the technical aspects of pilots (WP7) and the end-users, who rely on comprehensive documentation. This also includes assessing requirements and KPIs, as highlighted, in the final report of (WP8).

A. Open Callers' Questionnaire Responses

1) OC-Project title:
    - MOTION
    - MANTRA
    - IOTLORAMESH
    - Hopcast
    - HazardMiner
    - Poseidon
    - PROUD5G
    - CHEaaS

2) Have you used or are you planning to use the technical documentation provided here: https://assist-iot-enablers-documentation.readthedocs.io/en/latest/

(If you did not use it, the questions 3-5)

- MOTION: No
- MANTRA: Yes
- IOTLORAMESH: Yes
- Hopcast: Yes
- HazardMiner: Yes
- Poseidon:Yes
- PROUD5G: Yes
- CHEaaS: Yes

3) Does the documentation provides you with enough information to understand and utilize the enablers?

- MOTION: N/A
- MANTRA: 4
- IOTLORAMESH: 3
- Hopcast: 5
- HazardMiner: 4
- Poseidon: 3
- PROUD5G: 4
- CHEaaS: 4

4) Do you have any suggestions to improve or update the documentation?

- MOTION: N/A
- MANTRA: In some cases additional information about the deployment of the components could be improved.
- IOTLORAMESH: N/A
- Hopcast: N/A
- HazardMiner: N/A
- Poseidon: We plan to work on Federated Learning enablers. We would like to have more documentation related to the configurations required to have the different services connected together (if not readily available by installing the Helm packages on the same K8s cluster), more details on API payloads, and some procedures to test that the installation is working properly.

- PROUD5G: A more complete description of the Multi-link Enabler.
- CHEaaS:It is well structured. Some sections need to be completed but overall it is nice.

5) Do you find the documentation user-friendly?

- MOTION: N/A
- MANTRA: 4
- IOTLORAMESH: 2
- Hopcast: 5
- HazardMiner: 4
- Poseidon: 4
- PROUD5G: 5
- CHEaaS: 4

6) Which enablers do you plan to use?

- MOTION: OpenAPI Management enabler
- MANTRA: Semantic Repository enabler, Edge Data Broker enabler
- IOTLORAMESH: Edge Data Broker enabler
- Hopcast: Localization Tag, Smart Orchestrator, Edge Data Broker enabler, Long term data storage enabler, OpenAPI Management enabler, Location tracking enabler, Monitoring and Notifying enabler, Identity Manager enabler, Distributed broker enable
- HazardMiner: FL Orchestrator, FL Training Collector enabler, FL Repository enabler, FL Local Operations enabler
- Poseidon: FL Orchestrator, FL Repository enabler, FL Training Collector enabler, FL Local Operations enabler
- PROUD5G: Multi-link enabler
- CHEaaS: Location tracking enabler, Smart Orchestrator, Long term data storage enabler, Edge Data Broker enabler, Monitoring and Notifying enabler, Identity Manager enabler, Devices managment enabler, Enabler for registration and status of enablers

7) How good did you find the documentation in our deliverables when preparing your proposal?

- MOTION: The description of the data and the scanner setup for the Pilot3B is very good, with the exception of a few issues. Due to the large image size and the substantial number of images, it is not feasible to provide a solution that will run efficiently on a Jetson. To address this, we propose to provide a containerized OpenAPI enabler designed to run on Pilot3B's edge device which is very much the same and probably fits better to Pilot3B as it removes the communication overhead between the jetson and the edge node.
- MANTRA: It was OK
- IOTLORAMESH: The deliverables about the reference architecture were useful.
- Hopcast: Most of the information required was made available through the documentation. Discussions with partners helped clarify few things.
- HazardMiner: I found the documentation in your deliverables to be clear, comprehensive, and well-organized.
- Poseidon: I found it satisfactory.
- PROUD5G: We think deliverables were moderately useful to us. They did not contain all the aspects necessary for us to correctly interpret architecture.
- CHEaaS: They were good quality deliverables. However, there were quite high number of information. The online documentation, everything in one place is a much better option.

8) How different do you feel about the technical scope of ASSIST-IoT enablers and architecture comparing to now and when you were writing your proposal?

- MOTION: Now we have a much clearer understanding.
- MANTRA: It is the same
- IOTLORAMESH: I think the architecture contains a pretty complete list of components. It also contains the layers of the things to cloud computing continuum. It seems, however, that the enablers are implemented for those layers where sufficient hardware capacities are available to run Kubernetes components.
- Hopcast: Difficult to compare, it was almost a year ago. But the documentation was already helpful.
- HazardMiner: I believe that the technical scope of ASSIST-IoT enablers and architecture has evolved and improved since I wrote the proposal, demonstrating a positive development.
- Poseidon: Considering the Poseidon project scope, we expected the FL enablers to be ready at this point, but still one of them is still not ready and the project is very close to the end.
- PROUD5G: We misunderstood the functionality of enablers (Multi-Link enabler) when writing the proposal, which is why we had to make several corrections.
- CHEaaS: Our understanding about the ASSIST-IoT before and after the proposal remained nearly the same. However, after having interacted closer with the project's participants, we have now better understanding of it. But it still requires some further exploration. In particular, we lack some concrete use cases that we could work on.

9) How do you manage to fulfil any unexpected gap between the promised ASSIST-IoT technical scope in comparison what you have now available?

- MOTION: We constantly trying to meet ASSIT-IoT Technical scope. In terms of technology stack we are aware of what is needed but given the very specific nature of the Pilot we are facing some issues due to the massive data size.
- MANTRA: Contacting with the enablers providers.
- IOTLORAMESH: We regularly inform the ASSIST-IoT team about our progress and in our meetings we discuss options and solutions for any gap that may appear.
- Hopcast: For now, the only technical requirement we are missing is the ability to provide vehicle 2 vehicle direct communications. We fill the gap by emulating such a link with Android device 2 device link.
- HazardMiner: The technical documentation and instructions on git repo can be more comprehensive.
- Poseidon: We may need, for the FL scope, to focus more on theoretical documentation about the integration rather than actual implementation.
- PROUD5G: We had to adapt our scenarios to take into consideration our updated understanding of what the functionality that the enablers offer
- CHEaaS: We had some delays for obtaining some sample data, but it was not critical, we are progressing as planned. We may need to work on with the project team on some concrete use cases, the main ones and some alternatives in order to mitigate in terms of risks of not being able to fulfil the gaps for the main use cases.

10) Did you find the GitLab Repository useful?

- MOTION: 3
- MANTRA: 4
- IOTLORAMESH: 3
- Hopcast: 5

- HazardMiner: 3
- Poseidon: 4
- PROUD5G: 5
- CHEaaS: 3

11) Do you have any prior experience using and managing GitLab artifacts?

- MOTION: 3
- MANTRA: 4
- IOTLORAMESH: 3
- Hopcast: 3
- HazardMiner: 4
- Poseidon: 3
- PROUD5G: 5
- CHEaaS: 5

12) How would you rate the accessibility process at GitLab repositories?

- MOTION: 3
- MANTRA: 4
- IOTLORAMESH: 5
- Hopcast: 4
- HazardMiner: 4
- Poseidon: 5
- PROUD5G: 5
- CHEaaS:2

13) Based on your general experience using GitLab, would you have any recommendations to suggest?

- MOTION: Unfortunately, I do not have any access to ASSIST-IoT GitLab repositories.
- MANTRA: It is Ok
- IOTLORAMESH: GitLab as a repository is fine. Using the repository for also providing a higher availability of source code would have been useful, but here the license policies of the individual enabler providers play a role and a final decision may not be taken by them until the end of the project.
- Hopcast: Our git (more general than GitLab) expertise is not expert level. For now, we are facing no issue.
- HazardMiner: No
- Poseidon: No additional suggestions
- PROUD5G: We made a successful deployment using code provided on GitLab repository, although we had to overcome few bugs together with AssistIoT team. It also happened a few times that bugs appeared after installing a new version of the code. Perhaps performing deeper code tests before submitting changes to GitLab would prevent that situation. In addition introducing platform to submit bugs like Jira would streamline the debugging process.
- CHEaaS: No particular recommendation. Just a note that the access rights to the GitLab were not clear. We only have access to public repositories. We are told that we do not have access to other project level repositories.

References

[1.] ASSIST-IoT (2022). D6.3: Testing and integration plan - Final. Deliverable of the Horizon-2020 ASSIST-IoT project, Grant Agreement No. 957258.

[2.] ASSIST-IoT (2022). D6.7: Release and Distribution Plan - Final. Deliverable of the Horizon-2020 ASSIST-IoT project, Grant Agreement No. 957258.

[3.] ASSIST-IoT (2022). D6.6: Technical and Support Documentation - Final. Deliverable of the Horizon-2020 ASSIST-IoT project, Grant Agreement No. 957258.

[4.] ASSIST-IoT (2022). D6.2: Testing and integration plan - Initial. Deliverable of the Horizon-2020 ASSIST-IoT project, Grant Agreement No. 957258.

[5.] ASSIST-IoT (2021). D3.3: Use Cases Manual & Requirements and Business Analysis - Final. Deliverable of the Horizon-2020 ASSIST-IoT project, Grant Agreement No. 957258.

[6.] ASSIST-IoT (2022). D3.7 Architecture Definition – Final. Deliverable of the Horizon-2020 ASSIST-IoT project, Grant Agreement No. 957258