TALON

Autonomous and Self-organized Artificial Intelligent Orchestrator for a Greener Industry 4.0

Deliverable

D5.1 Installation & Demonstration Planning, Evaluation Methodology & KPIs Definition

Report

Actual submission date: 03/06/2024

Funded by
the European Union

Project Number:	101070181		
Project Acronym:	TALON		
Project Title:	Autonomous and Self-organized Artificial Intelligent Orchestrator for a Greener Industry 4.0		
Start date:	October 1st, 2022	**Duration:**	36 months
D5.1 Installation & Demonstration Planning, Evaluation Methodology & KPIs Definition Report			
Work Package:	WP5		
Lead partner:	INTRA		
Author(s):	Magda Foti (INTRA), Dorine Matzakou (INTRA), John Soldatos (INTRA)		
Reviewers:	Simon Sarkissian (8BELLS), Sergio Comella (ENG)		
Due date:	31/05/2024		
Deliverable Type:	R	**Dissemination Level:**	PU
Version number:	1.0		

Revision History

Version	Date	Author	Description
0.1	15/04/2024	INTRA	Table of Contents release
0.2	02/05/2024	All contributors	1st round of contributions about TALON components
0.3	02/05/2024	INTRA	Edits of the documents, formatting, quality control
0.4	09/05/2024	INTRA, All contributors	2nd round of partners' inputs/contributions
0.5	20/05/2024	INTRA	Preparation of the final version
0.6	22/05/2024	All contributors	3rd round of partners' inputs/contributions
0.7	28/05/2024	8BELLS, ENG	Internal review
0.8	29/05/2024	INTRA	Submit final version to the coordinator for QA
0.9	31/05/2024	MINDS, INTRA, DUTH	QA and finalisation of D5.1
1.0	03/06/2024	ENG	Final coordinator review before submission

Table of Contents

List of Figures

List of Tables

Definitions and acronyms

AI	Artificial Intelligence
ALM	Application Lifecycle Manager
APIs	Application Programming Interfaces
AR	Augmented Reality
BERT	Bidirectional Encoder Representations from Transformers
CA	Consortium Agreement
CPU	Central Processing Unit
CRF	Conditional Random Fields
CSV	Comma Separated Values
DNN	Deep Neural Networks
DoA	Description of Action
DRAM	Dynamic Random Access Memory
DSS	Decision Support System
E2C	Edge-to-cloud
EC	European Commission
ELMo	Embeddings from Language Models
EU	European Union
FL	Federated Learning
GA	Grant Agreement
GAN	Generative Adversarial Networks
GDPR	General Data Protection Regulation
GPT	Generative Pre-trained Transformer
HDD	Hard Disk Drive
IPFS	Inter Planetary File System
JSON	JavaScript Object Notation
K8s	Kubernetes
KPI	Key Performance Indicator
LBP	Local Binary Patterns
LLD	Low-Level Design
LSTM	Long-Short Term Memory
ML	Machine Learning
NER	Named Entity Recognition
NG-SDN	Next Generation – Software Defined Network
PII	Personal Identifiable Information
PPE	Personal Protection Equipment
PC	Project Coordinator
QoS	Quality of Service
RAM	Random Access Memory
RBAC	Role Based Access Control
SDLC	Software Development Lifecycle
SLO	Service Level Objectives
SOTA	State of the Art
TC	Technical Coordinator
TRL	Technology Readiness Level
UAT	User Acceptance Testing
UATV	Unmanned Aerial and Terrestrial Vehicles
UC	Use Case
UI	User Interface
VR	Virtual Reality
WP	Work Package
XAI	Explainable Artificial Intelligence

Disclaimer

Executive Summary

The vision of TALON is to design and develop next-generation industrial systems in terms of performance, adaptation, explainability, trustworthiness and transparency. TALON aims at sculpturing the road towards the next industrial revolution by developing a fully automated AI architecture capable of bringing intelligence near the edge in a flexible, adaptable, explainable, energy and data-efficient manner. In this direction, TALON researches and develops a rich set of components that enable AI-fuelled orchestration of resources across the edge-to-cloud continuum. Most importantly, the project integrates and demonstrates these components in several real-life use cases and demonstrators. The implementation of such demonstrators is a challenging task, as it requires the integration of diverse and technologically heterogeneous components, which asks for a structured planning of the integration work. Likewise, the evaluation of the demonstrators asks for a structured methodology that considers technological, integration and user-centred aspects, including relevant Key Performance Indicators (KPIs).

In this context, the present deliverable aims to provide a plan for the implementation and release of the various components of the TALON platform and demonstrators, along with an outlook and plan for their integration. Specifically, deliverable 5.1 “Installation & Demonstration Planning, Evaluation Methodology & KPIs Definition Report” provides a detailed description of the planning of the technical components’ implementation status during the two releases on month 22 and month 36 of the project along with the corresponding deployment/release schedule per component. Additionally, demonstration planning is provided for each use case: (i) Automatic UATVs Coordination, (ii) I5.0 Automation and Planning, (iii) AR/VR for Training and Maintenance and (iv) Human-Robot Collaboration, specifying the architectural modules that will be tested during the demonstration activities after each release. Finally, the evaluation methodology that will be employed to assess the effectiveness of the implemented system is presented together with its alignment with the TALON’s non-functional requirements. The present document provides the alignment of the technical deployment activities with the demonstration planning ensuring a robust and comprehensive validation process.

1 Introduction

1.1 Objective of the Deliverable

The main objective of this deliverable is to report the planning on how TALON's demonstrators will be carried out. Specifically, this document details the installation and demonstration planning for the four real-world use cases that validate TALON's functionalities across diverse industrial environments. It outlines the evaluation methodology that will be employed to assess the effectiveness of the implemented system, ensuring a robust and comprehensive validation process. Additionally, this deliverable reports on the status of key performance indicators (KPIs) definition that will be used to measure the success of the TALON system against its designated objectives.

1.2 Relation to other Work Packages

This is the first deliverable of WP5 (Integration, Validation and Demonstration) series of deliverables. The work reported in this document is part of *Task 5.1: Installation & Demonstration Planning, Evaluation Methodology & KPIs Definition*. The outputs of this deliverable will be used as part of *Task 2.3: Performance Assessment via Modelling & Real-World Results*, *Task 3.4: AI-based Resource Coordination Through Computation Offloading & Social Aware Caching*, *Task 3.5: Self-Healing & Self-Correcting Mechanisms*, *Task 3.6: Smart Pricing Policies for Non-Commercial Devices Participation*, *Task 4.2: Few-Shot & Hybrid Learning Approaches*, *Task 4.3: XAI, Monitoring & Visualisation mechanisms* and *Task 4.4: Security & Privacy Blockchain Mechanisms*.

1.3 Structure of the Document

This document is structured in 6 Chapters.

Chapter 1 provides an overall introduction to the deliverable objectives and positioning within the project.

Chapter 2 focuses on the installation planning of the three main layers, Access and Security, AI-fuelled Orchestration and AI Cognition. A detailed description of the planning of the technical components' installation is being reported along with a table of their corresponding deployment/release schedule.

Chapter 3 focuses on demonstration planning. This chapter delves into the specific plans for showcasing the four key use cases: Automatic UATVs Coordination, I5.0 Automation and Planning, AR/VR for Training and Maintenance and Human-Robot Collaboration. For each use case, the chapter describes the architectural modules that will be tested during the demonstration. It will also detail the specific demonstration planning for each use case, ensuring successful execution.

Chapter 4 outlines the evaluation methodology that will be employed to assess the effectiveness of the implemented system. This chapter introduces and describes the verification and validation model (V-model), which serves as the main pillar of the evaluation process. A mapping of TALON's non-functional requirements to the V-model is performed, ensuring a comprehensive evaluation.

Chapter 5 reports on the KPIs progress and actions.

Finally, **Chapter 6** concludes the deliverable along with the outlook.

2 Installation Planning

2.1 Access and Security

2.1.1 Authentication and Authorization

Status report and functionalities in M22

The authentication and authorization component supports two important functionalities that work together to secure access to end users, applications, Application Programming Interfaces (APIs), and resources. Figure 1 depicts the deployed component, as well as the landing page for the TALON end-users by prompting them to provide their credentials.

As far as the end users are concerned, the component has been fully deployed in M22 and supports user identity verification, credentials validation and session management. We have also deployed an API gateway which acts as a single-entry point for secure applications access. User identity verification involves the confirmation that only authenticated to the TALON platform users may access it using a username and password, which are considered as user credentials. The credentials are checked against a secure and hashed store for increased security to ensure they are valid. Last, the end user once authenticated, a session is then created to track their activity and grant access to the TALON system for a specific duration.

Regarding the applications and APIs, the authorization ensures that only authorized users can perform specific actions on resources, preventing unauthorized modifications or deletions. Also, it ensures access control which determines what resources a user can access based on their permissions. Permissions can be assigned based on roles (e.g., administrator, employer, user) or specific attributes. The component enforces defined access controls, restricting unauthorized access to sensitive data or functionalities.

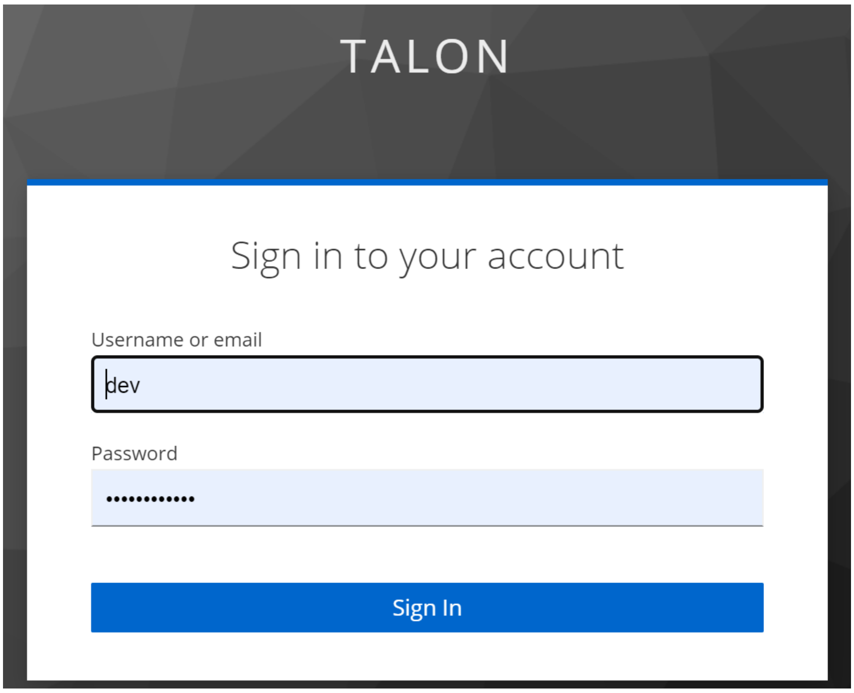

Figure 1: Landing page and user authentication in TALON

Status report, improvements and functionalities in M36

The authentication and authorisation component has been deployed in UBITECH's centralised cloud infrastructure. A dedicated server handles authentication and authorisation logic. This offers centralised control to both end users and applications. Currently, we support Role-Based Access Control (RBAC). RBAC assigns permissions based on pre-defined user roles (administrator, employer, employee). This simplifies access management and ensures that users only have the privileges they need for their tasks. This component is horizontal to the TALON pilots. By M36, we plan to further configure the component to support adaptive authentication if end users need it. This approach will enable to dynamically adjust authentication requirements based on attributes like device location, device type, and access time.

2.1.2 Data Anonymisation

Status report and functionalities in M22

Until M22, the text anonymisation module has completed the conceptualisation of the textual, numerical, and tabular anonymisation components and techniques. In the premise of the anonymisation of sensitive information and the TALON activities, the textual, numerical and tabular anonymisation component has been created that includes the functionality to anonymize various input data used by TALON subsystems for analytics, AI model training, decision support, process evaluation, and deployment-related operations. The data, mainly textual, tabular, or numerical, comes from various sources like reports, logs, sensor data, and edge devices. The developed anonymisation component was implemented to respect EU and international laws and regulations, implementing integrated and adaptable anonymisation techniques. This ensures that the system and its dependencies handle and process such data while respecting users' and data safety, security, privacy, and integrity.

In this period and for the development of this component, a variety of methods were employed and tested, both for recognising sensitive information in the form of Personal Identifiable Information (PII) and Named Entity Recognition (NER), utilising state-of-the-art and benchmark AI techniques. These techniques included Deep Neural Networks (DNNs), Conditional Random Fields (CRFs), Long-Short Term Memory models (LSTMs), Embeddings from Language Models (ELMo) and Transformers, for recognising the sensitive information that is subsequently anonymised. The anonymisation module includes different methods for obfuscating PIIs, namely, i) Removal, ii) Categorization, iii) Pseudonymisation, as it can also be seen in the following Table 1.

Table 1: Example of the anonymisation process

Methods	**Original Data**	**Transformed Data**
Removal	John Smith works at HSBC Bank	<REF> works at <REF>
Categorisation	John Smith works at HSBC Bank	<NAME> works at <LOCATION>
Pseudonymisation	John Smith works at HSBC Bank	Peter Green works at NatWest Bank

This process has been componentised, having provided a User Interface (UI) and integrated into the TALON dashboard. The dashboard provides the functionality to input raw text and anonymise it. Additionally, the functionality to input tabular data in the form of a .csv or Excel file is also being developed and under integration to the dashboard. Finally, all of this functionality is provided in the form of available APIs, available to the rest of the TALON components, offering on-demand anonymisation services.

Figure 2 below depicts the text anonymisation module integrated with the TALON dashboard.

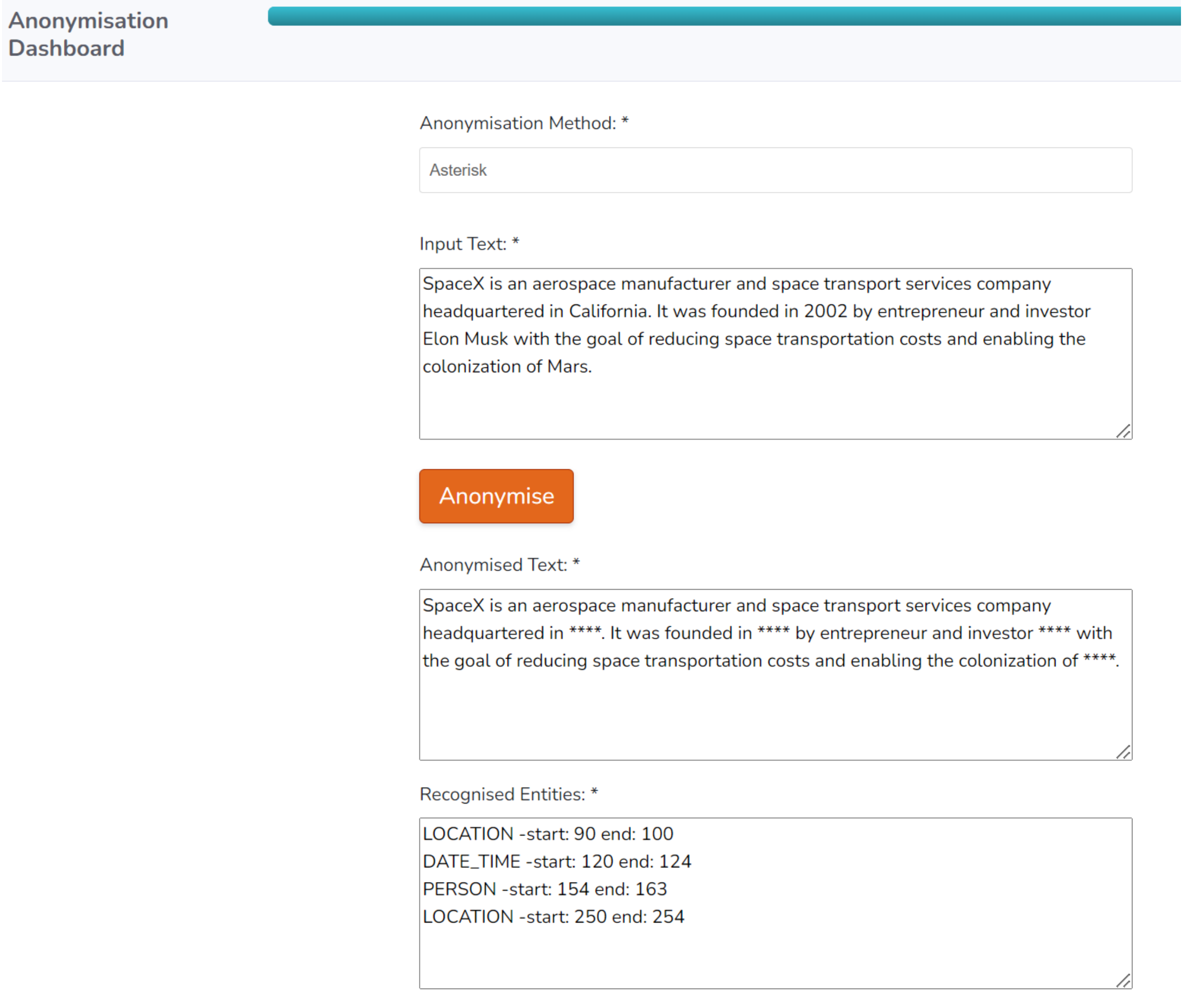

Figure 2: Anonymisation methods in TALON

The Image Anonymisation Module will be fully coded, and function tested (with public datasets) for M22.

The module can be configured as:

- *Anonymiser (original image/video cannot be retrieved)*
- *Pseudonymiser (original image/video can be retrieved by a secured MongoDB)*

The following functionalities are available, configurable and tuneable:

- *Face Detection for Image: Haar cascades, DNN, Local Binary Patterns (LBP), a combination of Haar and DNN*
- *Face Anonymisation for Image: Blurring, Gaussian Blurring, Pixelation, Black masking*
- *Face Detection for Video: Haar, DNN, LBP*
- *Face Anonymization for Video: Blurring, Gaussian Blurring, Pixelation, Black masking*
- *Exif-tags removal*

It is also available a cron-job auditor to delete, every day to a configurable hour, the original image/video files present in MongoDB whose dates are older than a configured retention period.

The following communication interfaces are available:

- *Image Upload and Processing*

  - *POST/upload*
- *Module Configuration, for setup, configure and tune the module*
  - *PUT/update_config*
  - *GET/retrieve_config*
- *GDPR Compliance*
  - *GET/retrieve_image_info_by_name*
  - *GET/retrieve_image_info_by_date*
  - *GET/download_image_id*
  - *GET/download_image_name*
  - *POST/delete_image*

Status report, improvements and functionalities in M36

For M36, the text anonymisation module will finalise the integration of additional functionality and integration with the rest of the TALON platform. In particular, the anonymisation component will fully integrate the functionality to support tabular data in the form of .csv/Excel files. Additional experiments will be employed to evaluate the efficacy of more advanced identification and obfuscation techniques, such as the employment of advanced models like Bidirectional Encoder Representations from Transformers (BERT), RoBERTa: A Robustly Optimized BERT Pretraining Approach, and Generative Pre-trained Transformer (GPT) models and Microsoft Presidio. Finally, the component will be employed in the related pilots to establish its utility in the TALON application scenarios.[3] models and Microsoft Presidio[4]. Finally, the component will be employed in the related pilots to establish its utility in the TALON application scenarios.

The Image Anonymisation module will be completed by M22, so no improvements or enhancements are foreseen for M36. Nevertheless, during the integration with the client sever if the anonymisation results will be considered inadequate, new face detection techniques will be analysed and if needed introduced. The Image Anonymisation Module is a key mechanism within the Access & Security layer designed to protect the confidentiality, integrity, and availability of data and systems. The module facilitates the protection of individuals' privacy captured in images used for AI development within the TALON project. By anonymising these data, the module ensures compliance with privacy regulations, safeguards sensitive information, and enables TALON to leverage visual data ethically and responsibly.

2.1.3 DLTs for Securing AI/ML models weights

Status report and functionalities in M22

For the TALON project, private blockchain-based mechanisms have been employed to provide an indispensable private, secure, and trustworthy landscape. Up until the first 22 months of the project all the basic functionalities will be developed according to the requirements. More specifically, a client application stores the AI/ML model weights received from the Federated Learning (FL) clients, on the blockchain network (Hyperledger fabric) utilising the fabric-gateway SDK. Each local model represents a digital asset in the blockchain environment.

Using client app we have the following communication interfaces:

- Asset management
  - POST/create_asset
  - POST/update_asset
  - POST/delete_asset
- Asset accesibility
  - GET/get_all_assets

  - GET/asset/:asset_id
  - GET/asset_history/:asset_id

Also providing an API for network/peer status and transaction/event indexing. Using indexer API, we have the following communication interfaces:

- Block accessibility
  - GET/blocks/:channel_id
  - GET/block/:channel_id
  - GET/blocks_by_date/:channel_id
- Transaction accessibility
  - GET/transactions/:channel_id
  - GET/transaction/:channel_id
  - GET/transaction_by_date/:channel_id
- Network/peer status
  - GET/transactions/:channel_id
  - GET/transaction/:channel_id
  - GET/transaction_by_date/:channel_id

The blockchain module. The client application receives input from various federated learning (FL) clients in JSON format and communicates this data to the Hyperledger Fabric blockchain network. The application initiates the process of transaction proposal, which continues through to the creation of a block. The integration status is pending in order to further finalise the implementation approach according to the reviewers' comments.

Status report, improvements and functionalities in M36

For M36 the module is expected to be fully integrated with all the up-and-running functionalities. The implementation roadmap involves the design and development of an IPFS based approach (instead of the existing on-chain), taking into consideration the reviewers' comments to evaluate further the optimal and most suitable approach for TALON's requirements. In more detail, a client app will store and pin AI/ML model weights on IPFS using Pinatas open-source node sdk. Pinata, as a pinning service, will fortify the availability of model weights on IPFS. The hash (CID) of AI/ML model weights will be stored in the blockchain for validity purposes as well as accessibility purposes. The hash of those data will be used as a URI for data availability.

2.2 AI-Fuelled Orchestration

2.2.1 Application Lifecycle Manager

Status report and functionalities in M22

Through M22, significant progress has been made in enhancing the Application Lifecycle Manager (ALM) to ensure seamless coordination, adaptability, and resource optimization across diverse application environments, including cloud and edge computing infrastructures. Real-time monitoring algorithms facilitate efficient deployment coordination, maintaining balance within the control loop while accommodating the autonomous nature of individual applications.
Built-in functionality enables dynamic enforcement of service level objectives (SLOs), triggering corrective actions when necessary, increasing ALM's flexibility to respond to changing conditions. Implemented capabilities maintain a unified view of resources across cloud and edge environments, optimising resource utilisation and ensuring fairness among competing workloads.

Figure

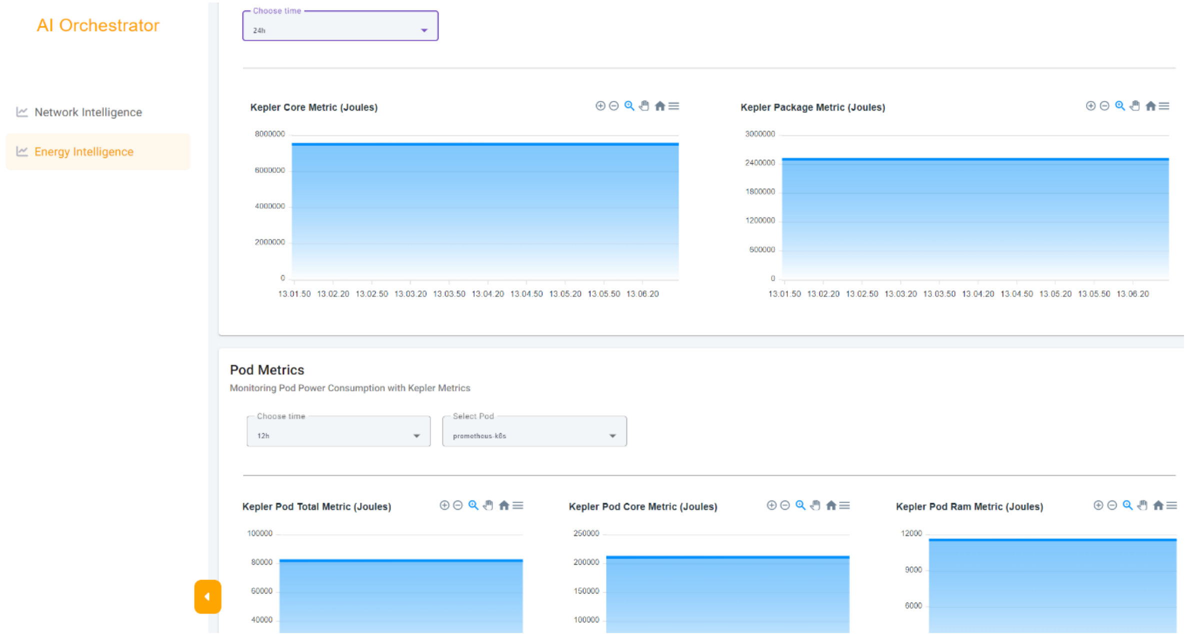

Figure 3: AI-fuelled Orchestration and Energy Monitoring

Status report, improvements and functionalities in M36

As part of our ongoing initiatives, we're committed to advancing the ALM by refining its capabilities. Our focus is on enabling scalability within the Edge-to-cloud (E2C) cluster. Our goal is to increase the versatility and deplorability of ALM, positioning it as a seamless component within diverse application environments. In addition, we will implement testing protocols to validate the ALM's adaptability to different deployment scenarios. Through testing, we aim to ensure the effectiveness and resilience of the ALM methodology, validating its effectiveness in covering key scenarios.

2.2.2 NG-SDN and Distributed Intelligence

Status report and functionalities in M22

Until M22, the Next Generation SDN (NG-SDN) and Distributed Intelligence component has been deployed and operates through a consistent network overlay among the edge and cloud nodes. It has been also linked with the four slices of operation that are coming from the Use Case applications. The tools of the deployment include the set up and configuration of a Kube-OVN SDN CNI for a K8s cluster. As for now the cluster has one (1) master and two (2) worker nodes. We have also defined a network topology along with the respective subnetworks. The network topology includes all the virtual routes and IP addresses to pods for inter-process communication. The figure below shows the descriptive analytics derived by harvesting the network monitoring metrics for an example edge-to-cloud application in the TALON dashboard. Last, we are collecting monitoring metrics from the utilisation of the network resources to support different Service Level Objectives (SLOs) and self-healing mechanisms. We are currently considering three (3) scenarios: (1) an edge streaming application where a policy should be enforced to increase its bandwidth for improved Quality of Service (QoS); (2) an overloaded gateway API where a policy is enforced for runtime adaptation and vertical scaling to its CPU and RAM resources; and (3) a suddenly terminated pod, where the Smart Policy Manager restarts the pod, creates a new virtual route and supports high availability of the end-to-end application only with a few moments of teardown.

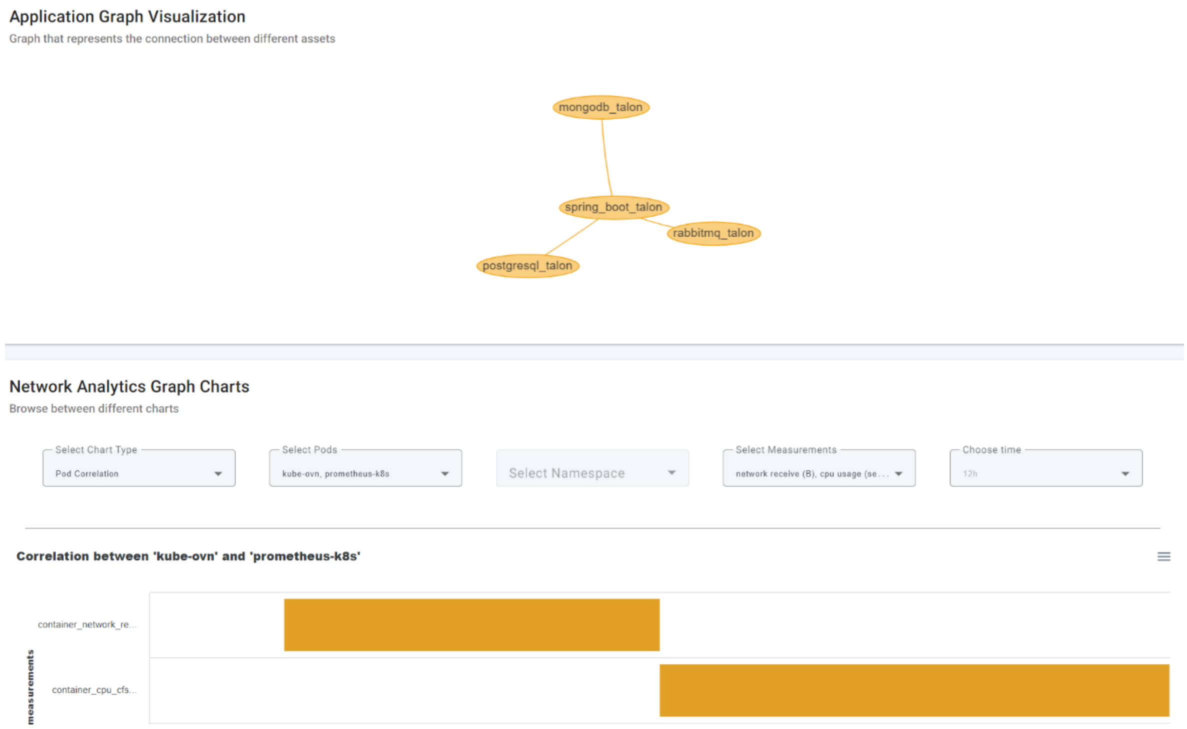

Figure 4: NG-SDN and Distributed Intelligence in TALON

Status report, improvements and functionalities in M36

Until M36, we plan to first migrate the distributed intelligence mechanisms to the pilot sites, collect their feedback and then apply any improvements on the policies definition or the optimisations / self-healing capabilities they are currently supporting.

2.2.3 AI Swarm Orchestration

Status report and functionalities in M22

PROBO has completed the basic offline orchestration code residing in the GCS and performed testing. The next step is to also finalize the development of the Cloud Orchestrator and integrate the communication among the systems We will need to operate a full-case scenario though, to test all the involved core partners and measure KPI and satisfaction.

The current functionality includes:

- Autonomous / self-localization per drone
- Smart distance based on node-to-node location
- Self-managed resource planning for each flight
- Off-board AI capabilities

Based on the above, certain requirements will likely change to better reflect the use cases after the PoC and pilots. In accordance with the functioning results and KPI achievements we will also further optimize the 3rd party integrated services and API.

Status report, improvements and functionalities in M36

Before the end of the project period, PROBO will have completed all tests, integrations with other partners and finalize the autonomous, “self-aware” management of resources based on the offline orchestrator with the option to use the online orchestrator (hybrid mode).

The sum of benefits will be:

- A hardware (drone) independent management solution for nearly all use cases. The management software will operate a variety of drone types. The application will be hardware agnostic so that, regardless of the use case and drone configuration, the pilot/operator can use it and react swiftly to all challenging demands.
- A low consumption - fast decision positioning for real-time swarm control. This enables a real-life and real-time scheduler that monitors vital signs of the drone such as: battery capacity, voltage, speed, altitude, connectivity, and board temperature with the aim to lower battery consumption and choose a better strategy for its next move. This is a complex process that also requires the feedback from extra sensors like cameras and ultra-sound.
- Autonomous or event-driven actions. This means autonomy level 5 where the pilot is basically a reviewer of the actions and keeps monitoring the performance of the system/platform. Of course, the operator/pilot can intervene and take control or any drone or the whole swarm at any time during the flying operation.

2.2.4 Resource Allocation and Deployment

Status report and functionalities in M22

By M22 we have employed various techniques to clean and prepare the dataset provided by Kepler for further analysis. This involved handling missing values through techniques like imputation or deletion and removing any duplicate rows to ensure data integrity. Subsequently, we generated summary statistics, offering insights into the dataset's statistical properties. Through the creation of various plots, histograms, scatter plots, and box plots, we aimed to unravel intricate data distributions and discern latent patterns or trends. This visual exploration provided us with invaluable insights into the inherent structure of the dataset, guiding our subsequent analytical endeavours. Moreover, we undertook a rigorous correlation analysis to unravel the intricate relationships between different variables within the dataset. Utilising correlation matrices and heatmaps, we explored the interplay between various metrics, shedding light on potential dependencies and associations. The understanding of correlations empowered us to identify key drivers and influencers within the dataset, laying the groundwork for the formulation of robust optimization policies. During the process of selecting datasets for optimisation of container energy within the Kepler environment, our focus was on aggregating and curating a comprehensive set of metrics that would provide a holistic view of energy consumption within containers. Each of these datasets was chosen to encapsulate key aspects of energy utilisation within containers, ranging from CPU usage and memory consumption to system and user CPU usage, as well as core, DRAM, and package-level energy consumption metrics. By aggregating these datasets, we created a unified dataset to be serve as the foundation for optimising energy efficiency within containerised environments. This dataset integrates resource utilisation and energy consumption data for each container, structured as time series data. The time series format of the dataset also enables tracking and modelling of energy consumption patterns and resource utilisation over time, providing a granular view of the dynamic interactions between different metrics.

Status report, improvements and functionalities in M36

Now we are focusing our efforts on data labelling, a critical step in the optimisation process, where we will assign labels corresponding to three distinct policies. These policies, namely scale up/down, data offloading, and task offloading, are integral to our goal of enhancing energy efficiency within the containerised environment. By accurately labelling the data according to these policies, we will pave the way for the development and implementation of targeted optimisation strategies.

Until M36, we will generate the labels for various policies such as scale up/down, task offloading, and load balancing. These labels will be integrated into the unified dataset and used to train machine learning models. The aim is to classify and recommend the most appropriate policy for optimising energy consumption based on real-time data. By associating specific energy and resource utilisation patterns with these predefined policies, the ML models can learn to predict the best actions to take, such as scaling resources up or down, offloading tasks to different containers or nodes, and balancing the load across the system.

2.2.5 Smart Policy Manager

Status report and functionalities in M22

The Smart Policy Manager offers functionalities to streamline policy management and enforce policies to optimise the Service Level Objectives (SLOs) of TALON linked with specific Key Performance Indicators (KPIs) and metrics. Until M22, we have collected metrics related to the Quality of Service (QoS), and usage of resources (e.g., CPU, RAM, energy consumption) at the network level. These metrics have been further analysed to support network intelligence and self-healing mechanisms. Those metrics have also been used to develop and deploy a fully containerized ML-driven smart pricing service that carries out the prediction of future user resource utilization requirements. In doing so, the target is to produce pricing and incentivization schemes for the users. At the same time, the smart pricing tool, being currently developed as part of Task 3.6, will enable transparency in pricing for the users, so as to ensure trust and facilitate efficiency, while bringing forward a greener edge computing network. Until M22 the bulk of work done has focused on testing various architectures of the APIs that will perform this service, along with prototyping the ML models that will facilitate the prediction of resource utilisation. The relevant pricing and the accompanying incentivization schemes were also tested in a development environment. This process was performed in an iterative fashion, aiming to identify the most suitable technical specifications required to build and deploy such a service that will exceed the current status.

We have deployed three (3) scenarios tackling different SLOs and enforcing different policies. We have also deployed the state engine to capture any difference or request from the application side. The first scenario assumes a streaming application (e.g., video from drone) with the requirement to allocate more bandwidth for improved QoS. In this case, the Smart Policy Manager alternates the YAML file and enforces the policy to allocate more bandwidth to the video streaming pod.

The second scenario assumes that a pod acting as the gateway API receives additional traffic linked with increased workload. When the CPU and RAM exceeds a threshold, the Smart Policy Manager vertically scales the pod of the gateway API by alternating the CPU and RAM allocation to greater values. The third scenario monitors the virtual routes of the cluster and its nodes as a whole using the networking capabilities. When a pod fails, the Smart Policy Manager defines new virtual routes, restarts the pod and assigns the new virtual route to the recovered pod. This ensures high availability to an application with minimum time of teardown.

Status report, improvements and functionalities in M36

Until M36, we plan to enrich the set of policies that are enforced by the Smart Policy Manager. These policies will further link orchestration decisions made by the TALON Orchestrator, with decisions about optimal placement made by the Resource Allocator. They will also enable the delivery of decisions that foster an adaptive, flexible and transparent pricing and an incentivization scheme aligned with the project's overall objectives, as well as about the energy consumption of the pods (e.g., AI Models, application pods, etc.)

2.2.6 Data Monitoring, Collection and Aggregation

Status report and functionalities in M22

Until M22, we have successfully deployed the mechanisms and the functions for continuously tracking and observing system, services and pods performance, application health, and energy consumption. This functionality acts as the Edge and Cloud Aggregator of the TALON system for all the metrics that are needed to support different Service Level Objectives (SLOs) and therefore optimisations. A unified technological stack has been adopted across all the partners who are responsible for populating metrics (e.g., Pilot providers) or for consuming metrics (e.g., TALON's component owners). This stack includes the Prometheus monitoring engine which interacts with the Kepler container power consumption engine which populates energy metrics. Last, for persistence over an extended period, we have deployed InfluxDB to collect and store all the required metrics from Prometheus. This component also includes the establishment of all the necessary data operations for querying and retrieving monitoring data.

Status report, improvements and functionalities in M36

In the upcoming period, we plan to enrich the monitoring metrics data collection with AI Models and pilot metrics collected in the context of training / inference and running the Use Cases applications.

2.2.7 AI Model Training, Adaptions and SLOs

Status report and functionalities in M22

Until M22, we have successfully implemented the three (3) scenarios linked with the Distributed Network Intelligence. In this context, we have harvested monitoring data at network level, and we have deployed the Smart Policies to heal, or vertically scale or support advanced QoS based on patterns learnt from this data. To achieve this, we are feeding a rule-based engine with patterns learnt after applying Machine Learning and Deep Learning methods. These methods either support a classification task such as workload classification and vertical scaling or a regression task to predict a specific value like in the increase of the QoS for bandwidth re-allocation.

Status report, improvements and functionalities in M36

Until M36, the functionalities of the AI Model Training for adaptation and SLOs achievement will be integrated with the AI-fuelled Orchestrator to derive some smart decisions based on historically learnt patterns. They will be also integrated with the Resource Manager to derive smart decisions about the optimal placement and allocation of the compute resources. Last, as a whole the AI Models Training will be validated within the pilot execution environments.

2.3 AI Cognition

2.3.1 Self-Healing and Self-Correcting

Status report and functionalities in M22

The self-healing and self-correcting mechanisms operate through a cycle of continuous monitoring, analysis, decision-making, and action. When a potential issue is detected, the system analyses its severity and potential impact on the network. Based on this analysis, appropriate actions are taken to mitigate the issue autonomously. These actions may include rerouting traffic, reallocating resources, or adjusting configurations. Furthermore, the system learns from its actions and outcomes, continuously improving its predictive capabilities and decision-making processes. This iterative learning loop ensures that the network becomes more resilient and adaptive over time. Additionally, human intervention is reserved for exceptional cases or situations that require manual intervention beyond the system's capabilities.

Developing a self-healing and self-correcting mechanism for an industrial network requires a tailored approach to address the unique challenges and requirements of industrial environments. Here's a suggested solution:

- **Comprehensive Data Collection:** Implement sensors and monitoring devices across critical points of the industrial network, including machinery, sensors, and control systems.
  Collect data on equipment performance, environmental conditions, energy consumption, and production metrics.
  Utilize protocols like OPC-UA (Open Platform Communications Unified Architecture) for interoperability and data exchange between different industrial devices.
- **Data Aggregation and Storage:** Aggregate the collected data in a centralized Data Pool within the industrial network infrastructure. Ensure data integrity and security by employing robust encryption and access controls.

The efforts until M22 were focused around the development of the self-healing mechanism using a rule-based mechanism. A Kubernetes cluster will be deployed at the UPV premises and the behaviour of the pods will be studied in order to create various failure scenarios together with the mitigation actions. This means that when there is a problem in a pod then automatically a mitigation action

Status report, improvements and functionalities in M36

The work that will be done by M36 will focus on the collection and the training of AI models based on real data coming from the TALON pilots.

- **Advanced Analytics and AI:** Implement AI-driven analytics to process the collected data and identify patterns, anomalies, and potential failures. Train AI models to recognize normal operating conditions and deviations that indicate impending failures. Leverage machine learning algorithms for predictive maintenance, enabling proactive identification of equipment issues before they cause disruptions. Utilize anomaly detection algorithms to identify cybersecurity threats and unauthorized access attempts.
- **Proactive Mechanisms:** Design proactive mechanisms to autonomously address identified issues and prevent potential failures:
- **Self-Healing:** Implement automatic fault detection and isolation mechanisms to identify malfunctioning equipment or components. Automatically reroute data flows or switch to redundant systems to maintain continuous operation.
- **Self-Correcting:** Automatically adjust equipment settings or parameters to optimize performance and prevent failures. Implement closed-loop control systems that adjust operations based on real-time data and predictive analytics.

2.3.2 AI Capabilities, Optimized and Transfer Learning

Status report and functionalities in M22

In the context of our ongoing project, considerable progress will be made in M22 towards the implementation of transfer learning techniques using cutting-edge algorithms. Our primary focus remains on enabling real-time object detection functionalities tailored for wearable devices, particularly Augmented Reality (AR) glasses. Among the array of algorithms explored, YOLOv8 emerges as the most prominent state-of-the-art (SOTA) solution. Distinguished by its single-stage object detection methodology, YOLOv8 has been meticulously engineered to deliver robust real-time object detection capabilities while simultaneously ensuring a minimal energy consumption footprint. In addition to this, Federated Learning (FL) has been implemented the frame of optimising AI model training, leveraging distributed data in a secure and private manner. FL incorporates the necessary functionality to utilise the AI models developed with talon to offer advanced cognitive functionality and utility to the TALON pilot and application scenarios.

Until M22, our efforts will be diligently directed towards refining the integration of transfer learning methodologies within the framework of our project objectives. Central to this endeavour is the seamless adaptation of YOLOv8, leveraging its advanced features to enhance the real-time object detection performance specifically tailored for wearable devices like AR glasses. By harnessing the power of transfer learning and the efficiency of YOLOv8, we are poised to deliver a sophisticated solution that not only meets but exceeds the stringent demands for real-time object detection in resource-constrained environments. On the Federated Learning front, in this period the initial implementation of Federated Learning has been developed, adapting on the specific preconditions set by the TALON pilots. A variety of AI models have been integrated to the TALON Federated Learning framework and tested in the problems-to-be-solved by the pilot scenarios, in particular, 1) regression and value forecasting in the industrial domain, like Remaining Utility Life (RUL and object detection and recognition. Table 2 provides some results of the FL evaluation calculations and realising real-time experiments. The figure below depicts the integrated federated learning capabilities by monitoring multiple experiments in the TALON dashboard.

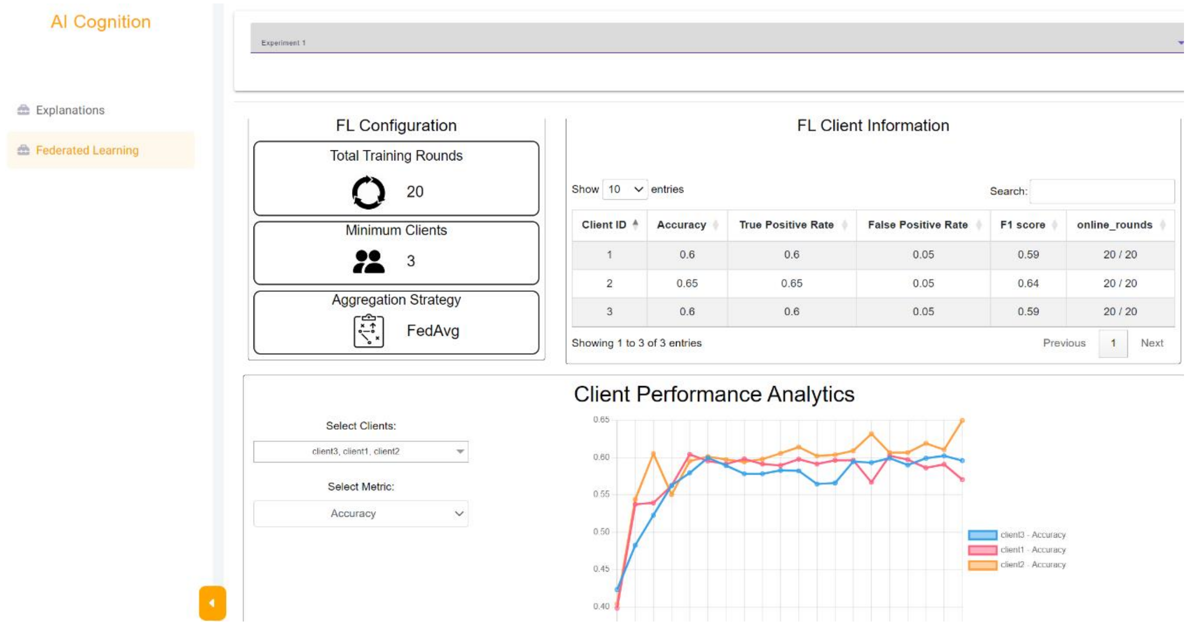

Figure 5: Federated Learning Capabilities in TALON

Status report, improvements and functionalities in M36

In our ongoing endeavours, the YOLOv8 model utilised for Transfer Learning is going to be refined and integrated: it is going to be containerised to seamlessly integrate into the orchestrator infrastructure. This move will ensure that YOLOv8 becomes readily deployable, primed to seamlessly function as an integral component within the orchestrator framework. Moreover, rigorous testing protocols are being implemented to validate the adaptability of this methodology across diverse usage scenarios. This meticulous testing process aims to ascertain the versatility and robustness of the methodology, ensuring its efficiency across a spectrum of use cases. In the premise of the FL, in the upcoming period, additional models and implementations are going to be developed and added to provide additional functionality to the TALON platform. Finally, final integration with the blockchain will be performed and tested to provide seamless and secure information flow.

2.3.3 Data Operations

Status report and functionalities in M22

Through M22, significant progress has been made with data enrichment. Specifically for the failure prediction use case, we utilized the MetroPT dataset. This open dataset is designed for predictive maintenance within the urban metro public transportation service in Porto, Portugal. It enables the prediction of failures in the metro system at least two hours in advance. Before going to the data enrichment task, we need to select the relevant features and discard the irrelevant ones. In this context, a comparative analysis of Information Gain Attribute Evaluation (IG) and Recursive Features Elimination (RFE) has been conducted.

Table 2: Classification performance with the whole dataset.

Model	Accuracy	Precision	Recall	F1-score
Decision Tree	98.14	78.37	79.91	79.12
Random Forest	**98.75**	**95.23**	**73.96**	**81.36**
Adaboost	97.78	53.56	35.57	36.81
XGBoost	98.26	91.27	60.76	68.11
Catboost	98.28	91.61	57.89	66.39
LightGBM	98.11	83.42	53.38	59.74

Table 3: Classification performance with the top 10 features selected by RFE

Model	Accuracy	Precision	Recall	F1-score
Decision Tree	98.14	78.39	79.77	79.06
Random Forest	**98.79**	**95.33**	**75.32**	**82.45**
Adaboost	97.78	53.56	35.57	36.81
XGBoost	98.26	91.27	60.41	67.92
Catboost	98.29	91.81	58.22	66.71
LightGBM	98.08	78.99	52.33	57.51

Table 4: Classification performance with top 15 features selected by IG

Model	Accuracy	Precision	Recall	F1-score
Decision Tree	97.92	75.85	78.06	76.92
Random Forest	**98.75**	**95.27**	**73.77**	**81.26**
Adaboost	97.78	53.56	35.57	36.81
XGBoost	98.26	91.26	60.54	67.97
Catboost	98.29	91.93	57.75	77.34
LightGBM	98.13	86.09	55.28	61.80

As the tables illustrate, the selected features demonstrate optimal classification performance. Leveraging these features, we generated new ones for data enrichment aimed at improving metrics such as the F1-score. This process involved transforming the timestamp column into granular metadata components: year, month, day, week of the year, day of the week, hour, minute, and second. We also seamlessly integrated environmental context by appending temperature and humidity readings corresponding to the geographical coordinates marked by the 'latitude' and 'longitude' features. Furthermore, the introduction of cyclical features for hours and months through sine and cosine transformations encapsulates the periodic nature of time.

The results (Figure 6) show significant improvements in F1-scores after data enrichment, especially for models like Decision Trees (DT), Random Forest (RF), XGBoost, CatBoost, and LGBM, which all reached near-perfect F1-scores.

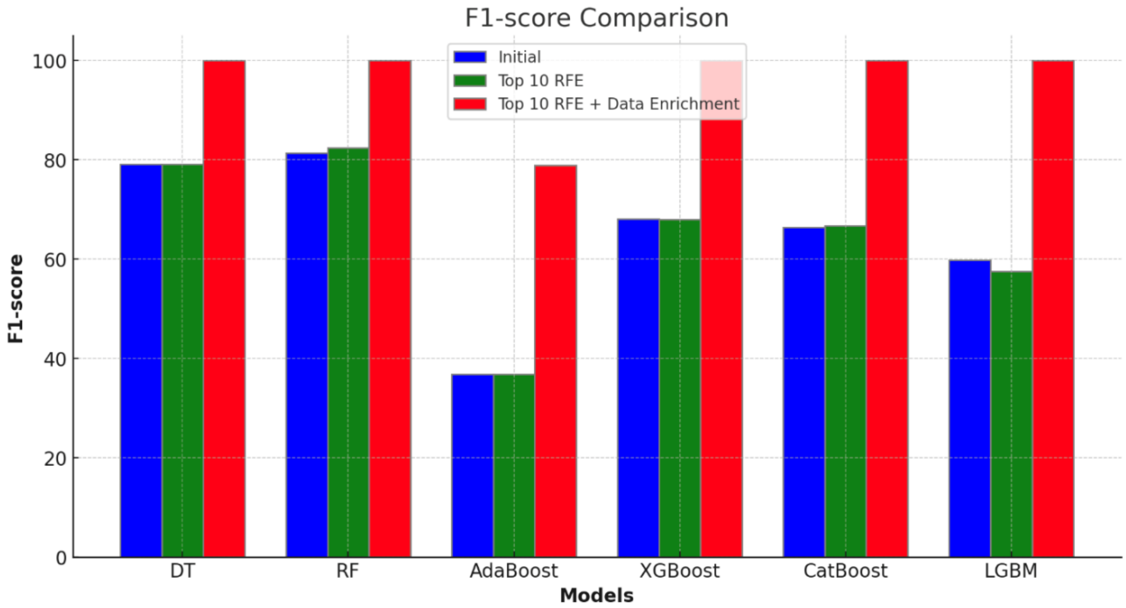

Figure 6: Failure prediction performance with the initial dataset vs. the top 10 features using RFE vs. and the top 10 features using RFE with data enrichment.

As part of our ongoing efforts for image enrichment tasks to improve fire detection capabilities using drone imagery, we have developed a Flask application that focuses on image augmentation and caption generation. This application is designed to interact with users through web forms, providing a straightforward interface for uploading and managing images.

- Image Enrichment process:
    - Augmentation: The application applies a range of transformations to the uploaded images to generate augmented versions. These transformations include flipping, rotation, and colour adjustment, which are essential for training robust machine learning model models.
    - Caption generation: Leveraging a pre-trained deep learning model, the applications generate descriptive captions for images. This feature helps to provide contextual understanding of the images, which can be useful for real-time monitoring and decision making in fire management.
- Basic configuration of the application:

- Upload Folder: This directory on the filesystem is designated for storing uploaded images, ensuring that user data is organized and easily accessible for processing.
- Augmented Folder: After images undergo augmentation, they are stored in this specific directory, allowing for easy retrieval and further analysis.
- JSON file: This file is used to store the generated captions, linking them back to their respective images for reference and use in further analytical work.

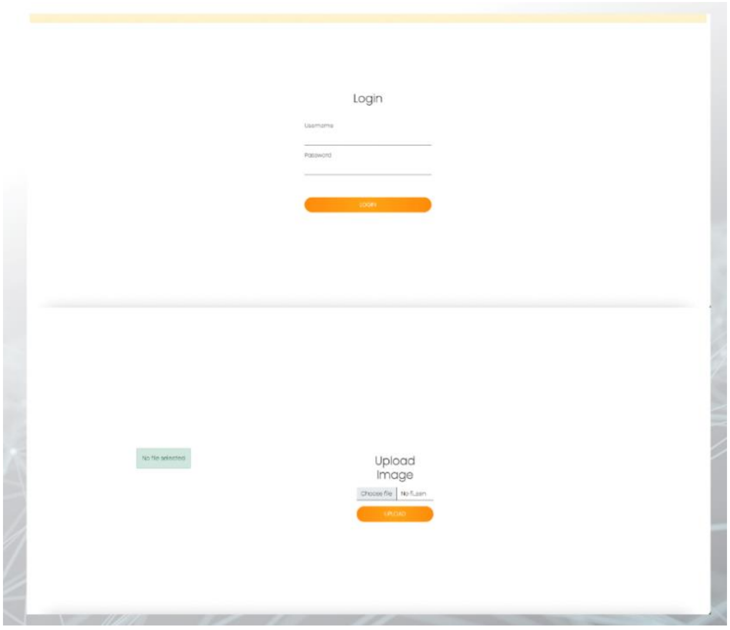

Figure 7: The login and upload interfaces.

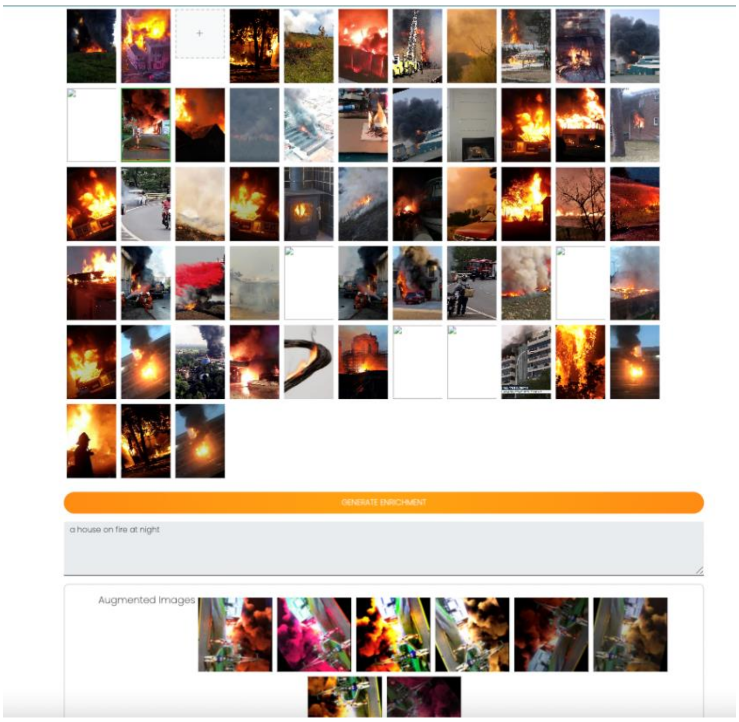

Figure 8: The display interface of the data augmentation and caption generation

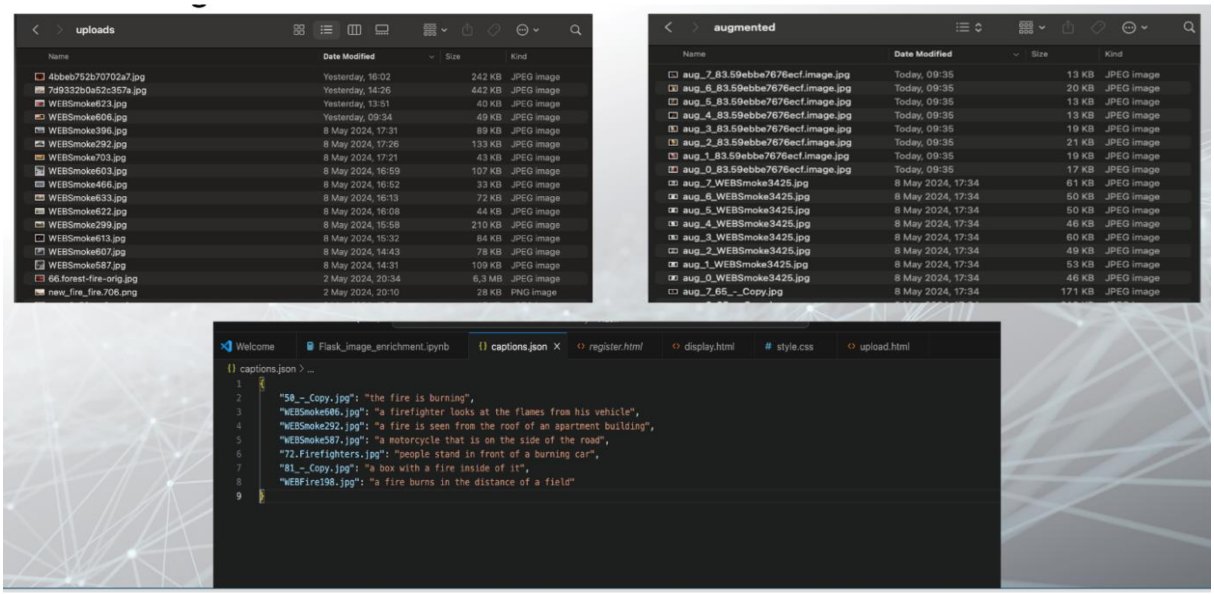

Figure 9: The display interface of the data augmentation and caption generation.

TALON's data operations also include a data curation module to facilitate curation tasks on incoming pilot datasets. The module supports many data curation operations that can be fine-tuned to meet the needs of TALON's pilots.
These operations include:

- Applicable on **tabular dataset**:
  - **Date Casting Algorithm**: This feature is particularly useful for standardizing date formats within tabular datasets. It allows for the transformation of dates from one format to another, ensuring consistency and compatibility across various data sources. For instance, if different datasets use different date formats, such as YYYY-MM-DD or MM/DD/YYYY, the date casting algorithm can preprocess them to a unified format, facilitating analysis and comparison.
  - **Winsorize:** Outlier removal is a critical step in data preprocessing, especially in statistical analysis where extreme values can skew results. The Winsorize algorithm, implemented with Scipy's scipy.stats.mstats.winsorize, identifies and limits extreme values, effectively reducing the impact of outliers on subsequent analyses. By statistically analysing the dataset, Winsorize helps maintain data integrity while minimizing the influence of outliers.
  - **Moving Window:** Outlier detection and removal in time series data are essential for ensuring accurate analysis and modelling. The moving window algorithm employs a Z-score calculated within a rolling window, enabling the identification of outliers based on their deviation from the mean or median of the data series. This method, commonly used in statistical analysis, provides a dynamic approach to outlier detection, adapting to changing data patterns over time.
  - **Time series interpolation**: Time series data often contain missing values or irregular intervals, which can impede analysis and modelling efforts. The time series interpolation functionality, leveraging Pandas' pandas.DataFrame.interpolate, addresses this challenge by filling in missing values based on neighbouring data points. With options such as linear interpolation, nearest neighbour, or polynomial interpolation, users can choose the most suitable method for their data, ensuring the continuity and integrity of the time series.
- Applicable on **object detection datasets**:
  - **Data augmentation**: Image data augmentation is a common technique used to increase the diversity and robustness of training datasets for object detection models. By applying

geometric transformations, colour manipulations, or other modifications to images, the data augmentation process generates additional training samples without requiring manual labelling. The integration of the Albumentations library enables efficient and customizable augmentation while preserving labelling information, enhancing the performance of object detection models.

- **Balance classes**: Class imbalance is a common challenge in object detection datasets, where certain classes may be underrepresented, leading to biased model performance. The balance classes operation addresses this issue by selectively augmenting specific classes, thereby increasing their representation in the dataset. By specifying the desired percentage or number of augmentations for each class, users can achieve a more balanced distribution, improving the model's ability to accurately detect objects across all classes.

The modular design of TALON's data curation module allows for flexible deployment options, ensuring optimal performance and resource utilisation. By executing curation operations close to the data source, unnecessary data transfer and processing overhead are minimised, streamlining the workflow and reducing latency. Each data curation algorithm can be independently deployed and configured according to the pilot's specific requirements.

The integration of the data curation module with the XAI module enhances TALON's data quality assurance capabilities. By leveraging explainable AI techniques, the XAI module identifies potential data imperfections and incongruencies, such as incomplete time series, unhealthy image datasets, outliers, or class imbalances. Upon detecting data issues, the XAI module can seamlessly communicate with the curation module to initiate remediation actions. Whether it's applying outlier removal algorithms, interpolating missing values, or augmenting underrepresented classes, the curation module addresses data quality issues in real-time, ensuring that the input data meets the desired standards for analysis and modelling.

Status report, improvements and functionalities in M36

Over the next months, our focus will be on advancing our data enrichment and image enrichment especially with the images for fire detection. For example, we will incorporate advanced image augmentation techniques using GANs (Generative Adversarial Networks) to create highly realistic synthetic images. Enhance the user interfaces and reduce the latency of the caption generation.

As far as the benefits offered per pilot from the data enrichment module are concerned, we plan to showcase the benefits of enrichment primarily in the execution of *Pilot 1: Automatic UATV Coordination* and *Pilot 2: I5.0 Automation & Planning*. In particular, regarding Pilot 1, images taken by the drones in the swarms would correspond to cars fires, and smoke that need to be detected. Hence, this relates to the test dataset provided by MINDS for fire detection for which work is ongoing for developing the corresponding image enrichment module. As far as the Pilot 2 is concerned, sensor-based time-series data, similar to the ones of the MetroPT dataset, will be originating from the Nakamura 2 machine, concerning, for instance, the power consumption in the turrets, temperature in the engines, room humidity, etc. These will be properly enriched so that AI-based failure prediction of the machine is enhanced.

2.3.4 Digital Twin

Status report and functionalities in M22

Substantial advancements have been achieved up to M22 in the development of the synthetic data generator solution, cantered around the TimeGAN architecture tailored for generating time series data. The synthetic data generator has been successfully trained on real-world CNC machine sensor data that first underwent the essential data preparation and pre-processing steps to ensure its quality and suitability for modelling. Qualitative assessments through PCA and t-SNE visualisation showed notable overlap between real and synthetic datasets, indicating faithful capture of structural characteristics. Quantitative analysis using KL divergence further supported this, suggesting a degree of similarity between the distributions of real and synthetic data. Performance evaluations, employing TRTR and TSTR methodologies, demonstrated the synthetic data's ability to replicate original patterns and achieve comparable training performance. Effort has been made in the containerisation of the solutions using Docker technology, to facilitate the deployment within the TALON Cloud-edge ecosystem and endure the portability and compatibility across diverse computing environments via a streamlined YAML configuration approach. Finally, additional data was systematically incorporated to fine-tune the model, aiming to bolster its ability to generalise across diverse machine conditions and manufacturing scenarios.

Status report, improvements and functionalities in M36

Continuing the advancement of the synthetic data generator solution involves several key future steps. Firstly, there's a need to refine the TimeGAN architecture, focusing on better capturing temporal dependencies and improving data generation accuracy. Additionally, expanding the training dataset's diversity and volume, potentially integrating more sensor data sources, or covering a wider range of machine types and scenarios, can enhance the model's robustness. Lastly, integrating with real-time systems and iterating based on user feedback will also be essential for practical deployment and ongoing improvement.

2.3.5 XAI and Monitoring

Status report and functionalities in M22

The figure below depicts the integration of Trust Level 1 and Trust Level 2 explanations regarding the data quality for different data modalities in the TALON Dashboard.

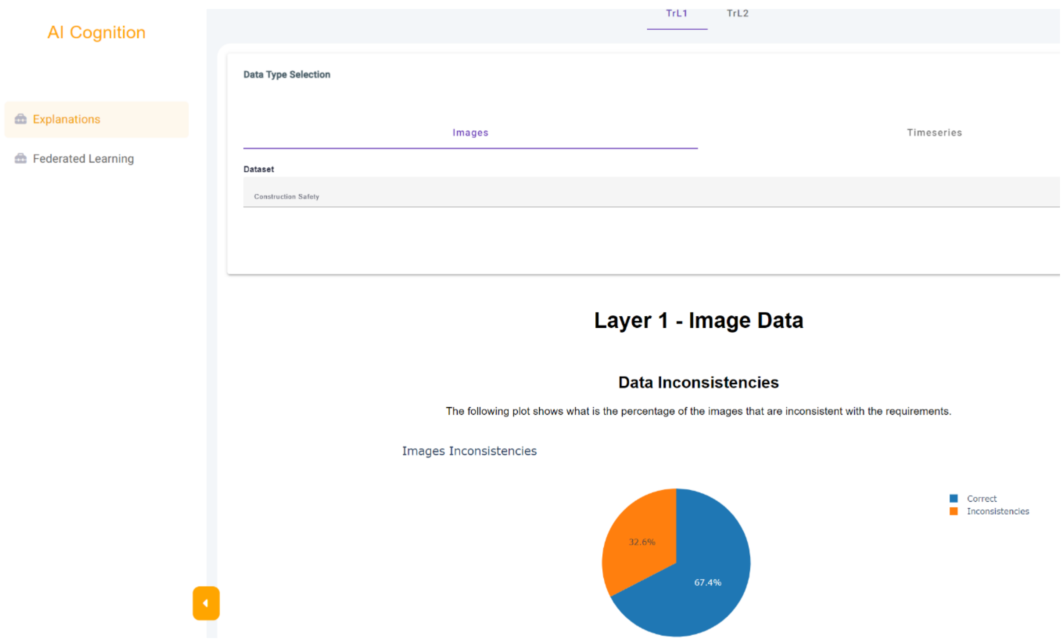

Figure 10: Explanation of TrL1 and TrL2 in TALON

Up until M22, the first two layers of the explainable AI framework (TRL1 and TRL2) have been developed and fully integrated within TALON platform, while TrL3 is developed but not yet integrated. Moreover, the XAI and Monitoring Module will be fully coded, and function tested (with public datasets) for TRL1 (for both images and timeseries), TRL2 (for both images and timeseries) and TRL3 (for images) for M22.

The module can be configured per data modality as:

- Images
  - Inconsistencies detector for TRL1 (based on image size, type etc.)
  - Misbalancing detector for TRL2 (based on the image assigned class)
  - Explainability provider for TRL3 (produced heatmaps for object detection task)
- Timeseries
  - Missing values detector for TRL1 (based on NaN values)
  - Anomaly/Outlier detector for TRL2

The following functionalities are available:

- Inconsistencies detector for Image: OpenCV library
- Misbalancing detector for Image: OpenCV library
- Explainability provider for Image: EigenCAM for Yolov8
- Missing values detector for Timeseries: Pandas library
- Outlier detector for Timeseries: LocalOutlierFactor

An implementation using FastAPI was performed for dynamical analysis.

The following communication interfaces are available:

- TRL1
  - POST/Inconsistencies detector for Images
    - Input: directory path, image_extensions (str), h_min (int), w_min (int), h_max (int), w_max (int)
    - Output: JSON response with dataset check results or error message
  - POST/Missing values detector for Timeseries
    - Input: CSV file, ts_feature_name (optional, default: "date")
    - Output: JSON response with null check results or error message
- TRL2
  - POST/ Misbalancing detector for Image
    - Input: CSV file
    - Output: JSON response with imbalance check results or error message
  - POST/ Outlier detector for Timeseries
    - Input: CSV file, ts_feature_name (optional, default: "date")
    - Output: JSON response with outlier check results or error message
- TRL3
  - POST/ Explainability provider for Image
    - Input: model_choice, file (image file)
    - Output: Streaming response with object detection XAI results or error message

Status report, improvements and functionalities in M36

All functionalities of all TRLs are expected to be developed and integrated before M35 which is the D4.3 final report. In more detail, all four TRLs will be developed and integrated to the platform for both data modalities, taking also into serious attention the reviewers' comments. TRL3 for time series SHAP/LIME algorithms will provide a model agnostic explainability approach, while CAM algorithms will highlight the most significant parts of the images, while TRL4 for time series is under investigation. Finally, the fidelity and consistency of the AI models related with image data at TRL4 will be assessed, to check for potential limitations and evaluate the overall robustness.

2.3.6 Polyglot Data Management

Status report and functionalities in M22

Until M22, we have successfully deployed the different databases for the Polyglot Data Management. These have been centralised in the core backend services of the TALON system to efficiently manage different types of data and leverage the strengths of different databases for specific tasks. We are currently supporting different data modalities through the MinIO object store, MongoDB reporting system and PostgreSQL relational database. This approach enables us to combine operational data (e.g., transaction data) stored in a relational database with behaviour data stored in a NoSQL database. It also enables to store large unstructured or binary data in the centralised cloud while keeping structured data in the relational database. We have also deployed a mediator-wrapper to ease the querying and access across different storage systems.

Status report, improvements and functionalities in M36

In the upcoming period and until M36, we will enrich with additional business data the Polyglot Data Management and we will deploy and configure the common AI Models repository.

2.3.7 Visualisation Dashboard

Status report and functionalities in M22

Until M22, the Visualization Dashboard of TALON has both served as a digital tool to demonstrate and transform complex data sets into actionable insights and as a unified interface integrating different functionalities and components of TALON. As for now, it provides monitoring and tracking capabilities from network and energy metrics of edge-to-cloud pods in real-time. This helps identify peaks, trends, spot potential issues quickly and track the evolution of pods. It also helps to improve the user experience, as many TALON components are being executed in the background, so this approach assists end users to drill down into specific data points, filter information, gain specific insights and customise their view. The figure below shows the high-level view of the different functionalities of the Visualisation Dashboard. Functionalities which support the AI Orchestrator include the network and energy descriptive analytics. The functionalities of the AI Cognition cover the Explanations (XAI) and the conduct of Federated Learning experiments. Last, regarding secure access, we have integrated the authentication mechanism and we have integrated the textual anonymisation functionalities. The paragraph that follows reports on our plans regarding the improvement and evolution of the TALON Visualization Dashboard.

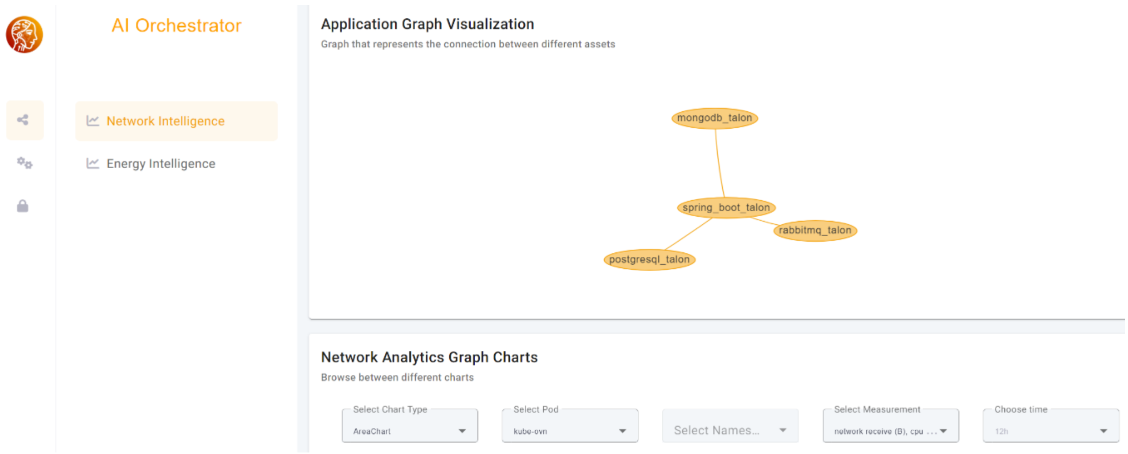

Figure 11: TALON's Visualization Dashboard

Status report, improvements and functionalities in M36

Until M36, we plan to improve the Visualisation Dashboard by collecting feedback by the end users. Also, we plan to link metrics with SLOs and KPIs to validate the TALON project objectives and the objective per Pilot / Use Case. The user journey will be further improved with filters and controls letting end users change timeframes and adjust settings to get a more focused view of the data. Last, by revisiting the requirements and user needs we plan to deploy reporting mechanisms to let end users collect notifications and alerts when specific conditions are met or violated, prompting users to take action and keeping the human-in-the-loop.

2.4 Technical Components Release and Deployment Schedule

Summarizing the previous paragraphs Figure 12 presents the whole TALON architecture while Table 5 provides the summary of the release and deployment plan per architectural component.

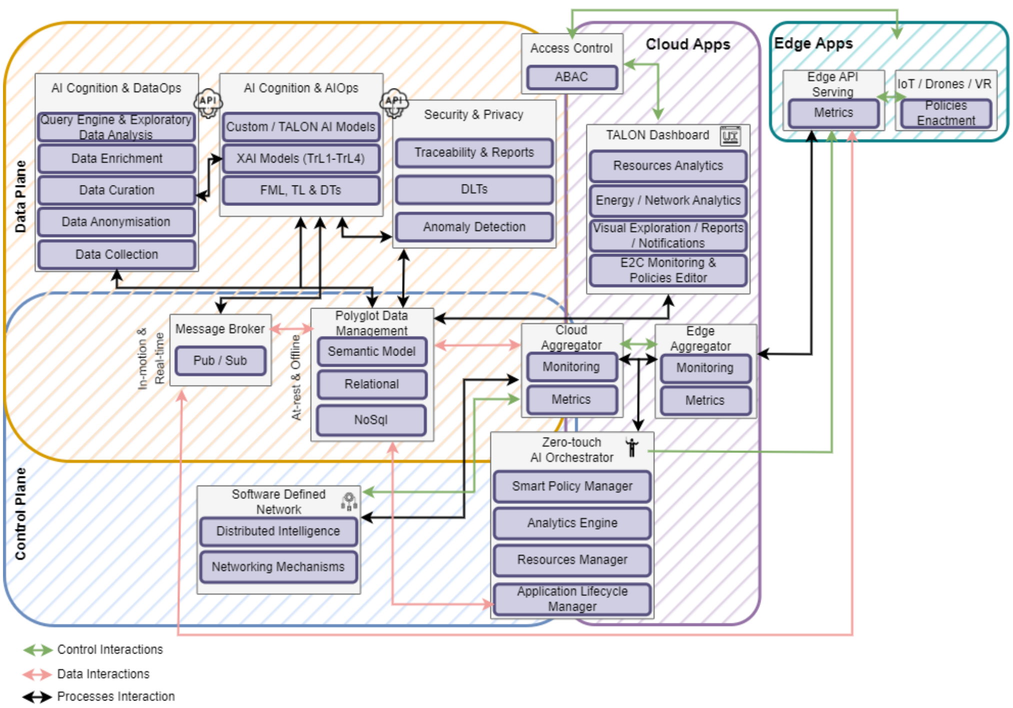

Figure 12: TALON Architecture

Table 5: Technical components release and deployment.

	Release 1 (M22)	Final Release (M36)	
Access and Security Layer			
Authentication and Authorization	Fully Deployed	Fully Deployed	Options to be filled
Data Anonymization	To be developed	Fully Deployed	To be developed
DLTs for Securing AI/ML models weights	To be developed	Fully Deployed	Partially Deployed
AI-fueled Orchestration Layer			
Application Lifecycle Manager	Partially Deployed	Fully Deployed	
NG-SDN and Distributed Intelligence	Fully Deployed	Fully Deployed	
AI Swarm Orchestration	Partially Deployed	Fully Deployed	
Resource Allocation and Deployment	Partially Deployed	Fully Deployed	
Smart Policy Manager	Partially Deployed	Fully Deployed	
Data Monitoring, Collection and Aggregation	Fully Deployed	Fully Deployed	
AI Model Training, Adaptions and SLOs	To be developed	Fully Deployed	
AI Cognition Layer			
Self-healing and Self-correcting	To be developed	Fully Deployed	
AI Capabilities, Optimized and Transfer Learning	Partially Deployed	Fully Deployed	
Digital Twin	Partially Deployed	Fully Deployed	
XAI and Monitoring	Partially Deployed	Fully Deployed	
Polyglot Data Management	Partially Deployed	Fully Deployed	
Visualisation Dashboard	Partially Deployed	Fully Deployed	

3 Demonstration Planning

3.1 UC1: Automatic UATVs Coordination

3.1.1 Architectural Modules to be Tested

The Architectural Modules to be tested and the most important developments are described below.

- Resource Manager
  - Monitoring Module
    Allocation of CPU, RAM and HDD
  - Efficiency Module
    Logging for the KPI
- Deployment Management
  - Communication Module
    This is the module that connects each drone with the GCS. The communication module consists of Wi-Fi and optionally a 4G/5G submodule. The configuration enables either local or remote connectivity, In the PoC and pilots only the Wi-Fi modules will be tested.
  - DSS Module
    This is the module that applies rules for decision support. The DSS module has a predefined set of logical path rules that are applied based on actions taken by the pilot and the drone in the warm.
  - Position Module
    This module sets 3D positioning of each drone. The position module observes the drone's relative position to other drones and actual position of itself with its internal GNSS. It keeps the swarm formed and consistent.
  - CA Module
    This is the collision avoidance module that defines which neighbouring drones are in a collision course. It interrupts the drone movements and instructs the pilot or the drone (in fully autonomous mode) to alter its course.
  - Inter-Com Module
    The intercommunication module for drone-to-drone info exchange is very important as it provides a common channel with which any drone can "speak" to each other. Since the drone we will be using is not custom-made the Inter-Com module will be a pure software interface inside the GCS. Otherwise, it would have been a drone-to-drone module on each drone body.

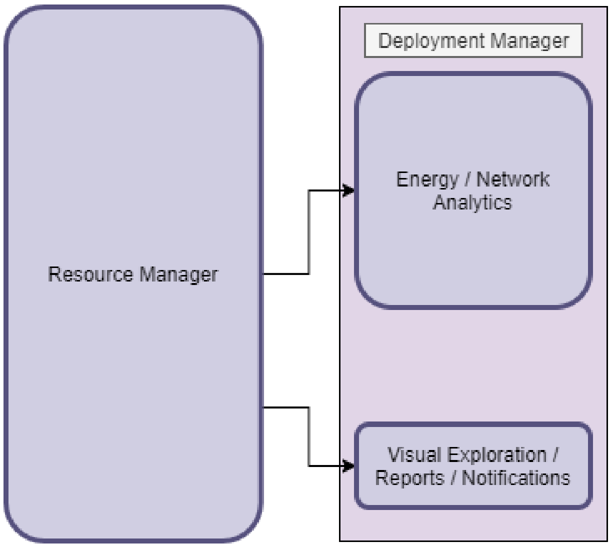

Figure 13: Architectural Modules to be tested.

3.1.2 Demonstration Planning

The planned modules per month will be the following:

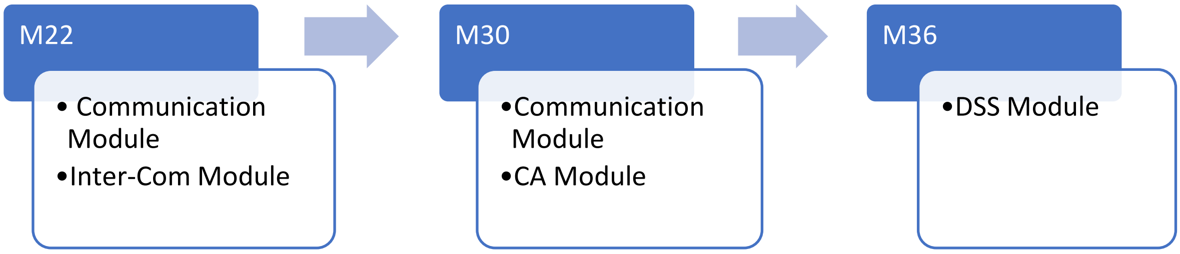

Figure 14: PROBO UC1 Demonstration planning

3.2 UC2: I5.0 Automation and Planning

3.2.1 Architectural Modules to be Tested

The Architectural Modules to be tested and the most important specific developments of each one are described below by groups, always from their functional point of view, as can be seen in the following Figure 15:

- Edge API Serving
- Access Control
- Security & Privacy
- AI Cognition & DataOps
- AI Cognition & AIOps
- TALON Dashboard
- Cloud Aggregator

- Zero-touch AI Orchestrator

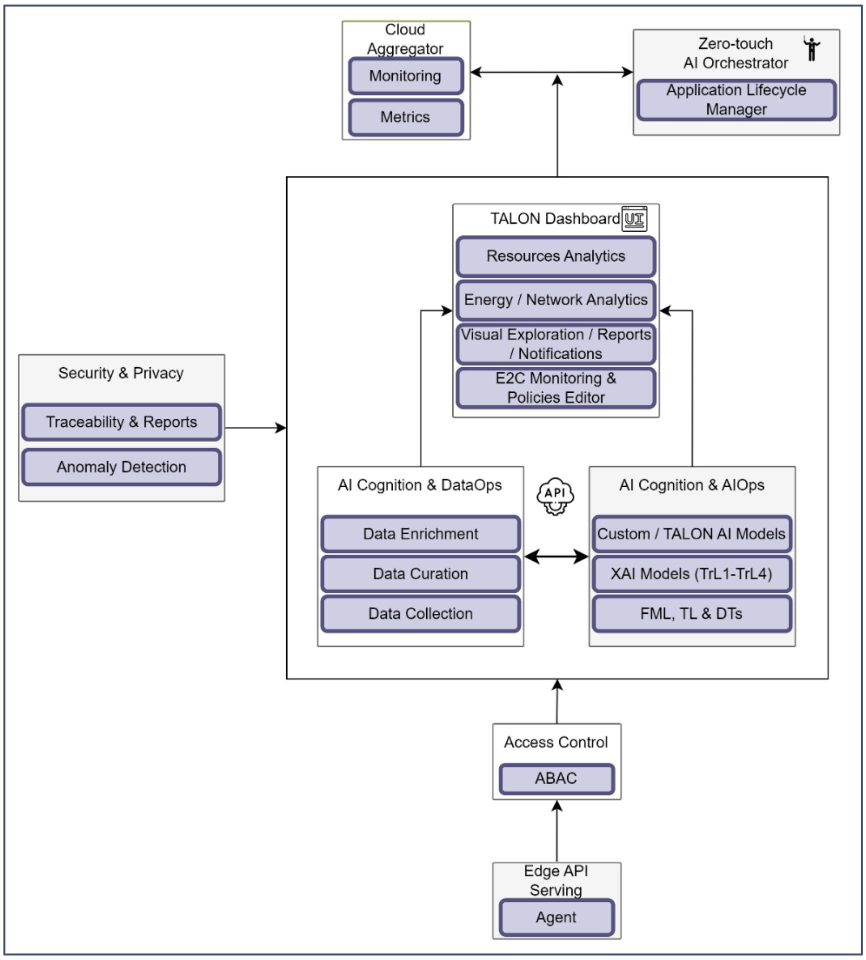

Figure 15: FACTOR Architectural Modules to be Tested

3.2.2 Demonstration Planning

The planned deployment schedule of the TALON modules is as follows, taking into account the development status of the TALON solutions and the adaptability to the FACTOR User Case.

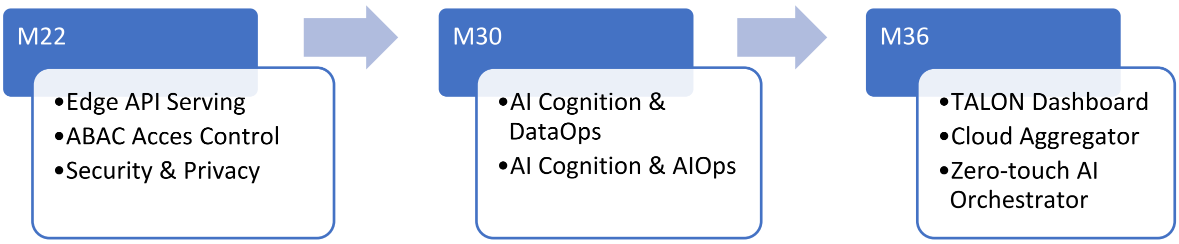

Figure 16: FACTOR UC2 Demonstration Planning

The sequence and order of execution will be subject to the progress and maturity of the development of the TALON modules and the adaptation capacity of FACTOR's production systems, always seeking that the result of the deployment is optimal.

Regarding scenario 2, the goal is to help the Network Engineering in the root cause analysis of the optical fiber link failure by monitoring and classifying different loss of signal scenarios with the use of explainable AI techniques. Exploiting ML driven classification simplifies the Network Engineering trouble-shooting task allowing to take a proper action much faster with great cost saving for the infrastructure owner. Besides the fault classification itself, predicted by ML, it is important to add explanatory information to support the decision-making process and select the proper fixing or investigating actions since the classification explanation can complement the root cause analysis with the necessary reliability, trust, and transparency required to reduce the risk of misinterpretation.

For the Proof of Concept of UC2 scenario 2, the TEI dataset that consists of a collection of time-series, recording the received optical power during an optical link fault, is used and can be assumed that these time-series were coming from physical HW such as machines, sensors and computers interconnected in the factory.

Significant progress has been made up to M22: the optical power loss time-series were subjected to a filtering process to ensure their quality and relevance for subsequent analyses; from a larger set of files have been identified and retained in the dataset the time-series that met specific validation criteria in the optical domain. Subsequently, data analysis, preprocessing and ad hoc time-series transformation has been performed; more ML classifier for fiber fault classification has been experimented to predict different types of faults in optical fiber link as well as XAI models for their interpretation.

Considerable result has been also obtained in the research field.

The Architectural components activated in UC2 second scenario are the “AI Cognition & Data Ops” for the data preprocessing and “AI Cognition & AI Ops” to train and assess the performance of one or more supervised machine learning and XAI model(s) developed by technical partners. The results are viewable through the TALON Dashboard.

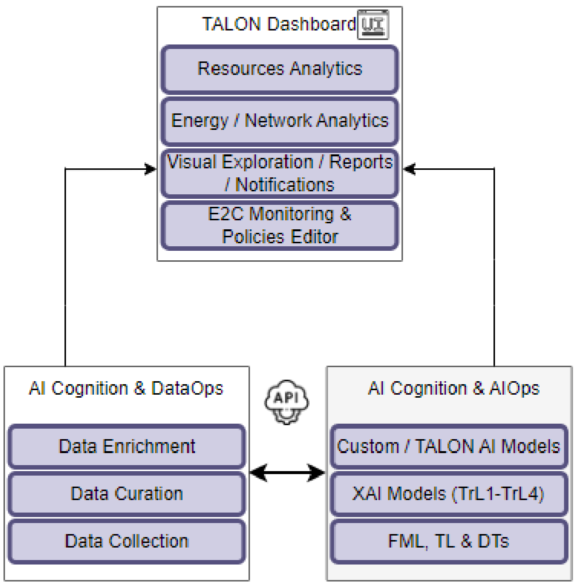

Figure 17: UC2 second scenario Architectural Modules to be tested

3.3 UC3: AR/VR for Training and Maintenance

3.3.1 Architectural Modules to be tested

AR Maintenance Application

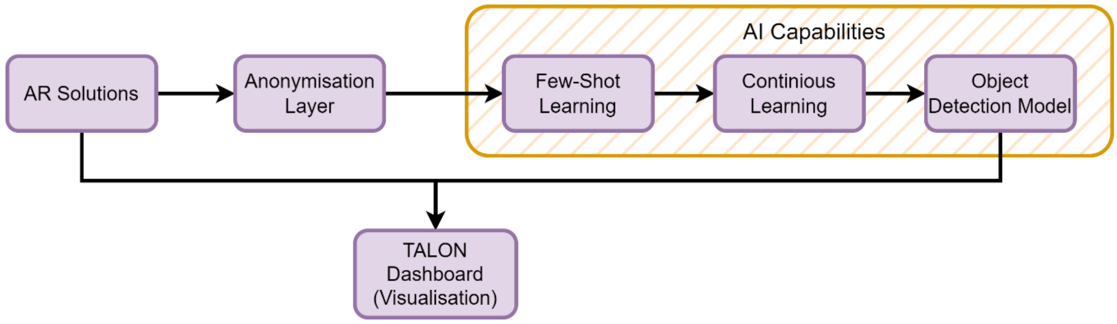

Figure 18: Overview of AR-based Maintenance Application

The application exemplifies a communication solution designed for Maintenance and Support personnel within an industrial context, augmented with AI models for scene analysis. This solution facilitates multi-modal communication, thereby expediting the maintenance process, reducing costs, and minimising downtime.

In compliance with current and anticipated future regulations such as the EU AI Act and GDPR guidelines, the application incorporates the Anonymisation Layer of the TALON platform. This layer enables the detection and removal of sensitive information from visual data, while preserving relevant information essential for scene analysis, such as object detection.

As part of its AI capabilities, the application employs both few-shot learning and continuous learning solutions. To enhance object detection, the AR Maintenance pilot allows users to add detectable objects, and the few-shot learning solution enriches the provided images, thereby diversifying the training data and improving results. Additionally, the continuous learning approach permits the dynamic addition of new data to existing models, enhancing capabilities with minimal resource usage.

To monitor the processes within the solution, we utilize the TALON dashboard. This dashboard offers a visual representation of the data generated by personnel communication, including hardware resource usage, AI model processing speed, network usage, and other relevant metrics.

VR training application

Figure 19: Modules Architecture Usage

The forthcoming VR training application is set to leverage the Polyglot Data Management system, see Figure 19, for the storage and management of training-related information. Within this framework, a bespoke account category named "Instructor" will be established. This dedicated account type will serve as a pivotal conduit for instructors to seamlessly upload and manage trainee-related data, including but not limited to their performance scores and other pertinent metrics. This streamlined process ensures efficient data handling and facilitates the seamless integration of instructor-led assessments within the VR training ecosystem.

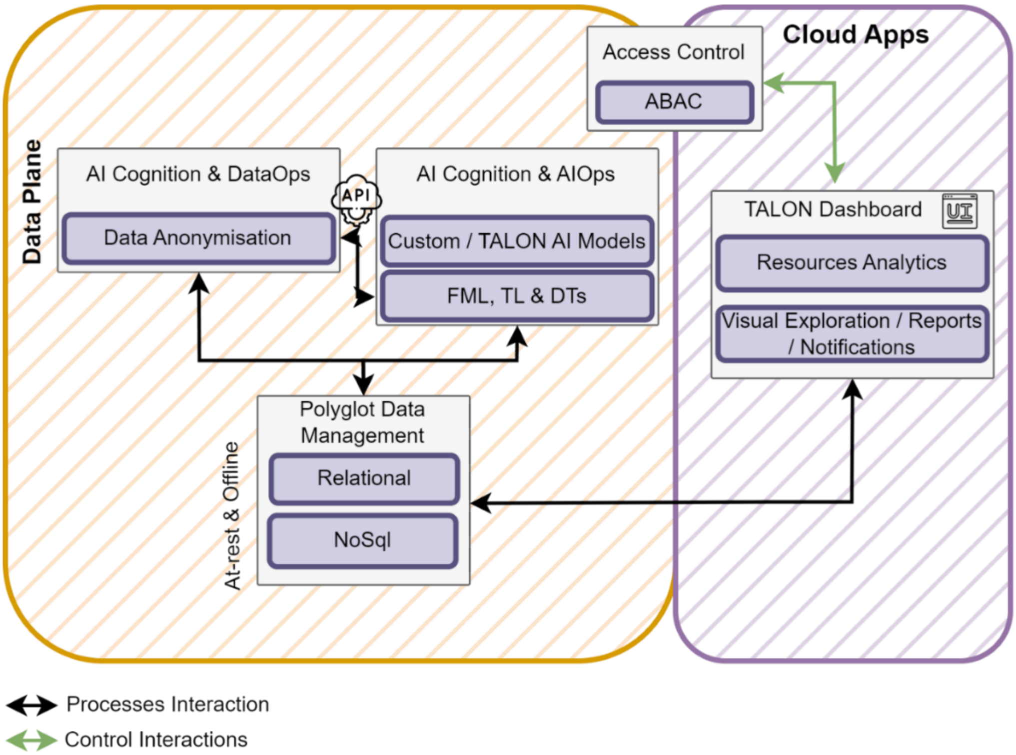

Figure 20: UC3 Architectural Modules to be Tested

The integration and interaction between the VR training application and the TALON platform will be managed through a robust Authentication and Authorisation module, as depicted in Figure 19. This critical component serves as the cornerstone for establishing a secure connection between the two systems, prioritising data integrity and confidentiality. Through stringent authentication protocols and granular authorization mechanisms, this module not only safeguards the connection between the VR training application and the TALON platform but also effectively manages associated risks, mitigating potential vulnerabilities and ensuring seamless interoperability between the two systems. The Figure 20 summarises the modules of the TALON architecture that will be tested in this use case.

3.3.2 Demonstration Planning

The demonstration for both the AR Maintenance and VR Training applications will take place at the FACTOR premises. This demonstration aims to compare the current solution with the solutions provided by the pilot applications. Multiple participants will be involved in demonstrating both approaches to ensure a fair comparison. Throughout and following the demonstrations, qualitative and quantitative data will be collected to assess the benefits of the approaches developed for the use case.

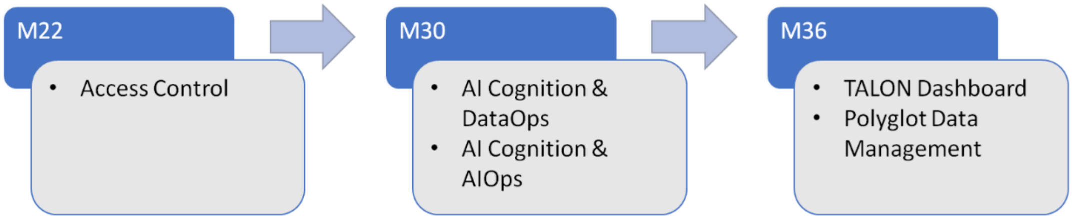

Figure 21: UC3 Demonstration Planning

The development of the tools that will be used in UC3, are in progress. The Figure 21 illustrates modules that will be ready for demonstration on M22, M30 and M36 by the end of the project. The demonstration will include detailed presentations of the functionalities and features of the AR Maintenance and VR Training applications. This will provide attendees with a comprehensive understanding of how these technologies can enhance maintenance processes and training methods within an industrial context.

To ensure objectivity and accuracy in data collection, standardised metrics will be utilised, focusing on factors such as efficiency gains, error reduction, and user satisfaction. These metrics will allow for a thorough evaluation of the effectiveness of the pilot applications in addressing the identified use case challenges.

3.4 UC4: Human-Robot Collaboration

3.4.1 Architectural Modules to be Tested

The Architectural Modules to be tested and the most important developments are described below by groups. The illustration of these architectural modules is also presented in the following Figure 22.

- AI Cognition & DataOps:
  - Query Engine and Exploratory Data Analysis (UI)
  - Data Collection
  - Data Anonymisation
  - Data Curation
  - Data Augmentation (for images)
- AI Cognition & AIOps:
  - Custom/TALON AI Models (pod logged metrics)
  - XAI Models
- Security & Privacy
  - Traceability and Reports (human interaction)
- Access Control
  - ABAC
- TALON Dashboard
  - Resource Allocation
- Cloud Aggregator
- Polyglot Data Management
- Zero-Touch AI Orchestrator

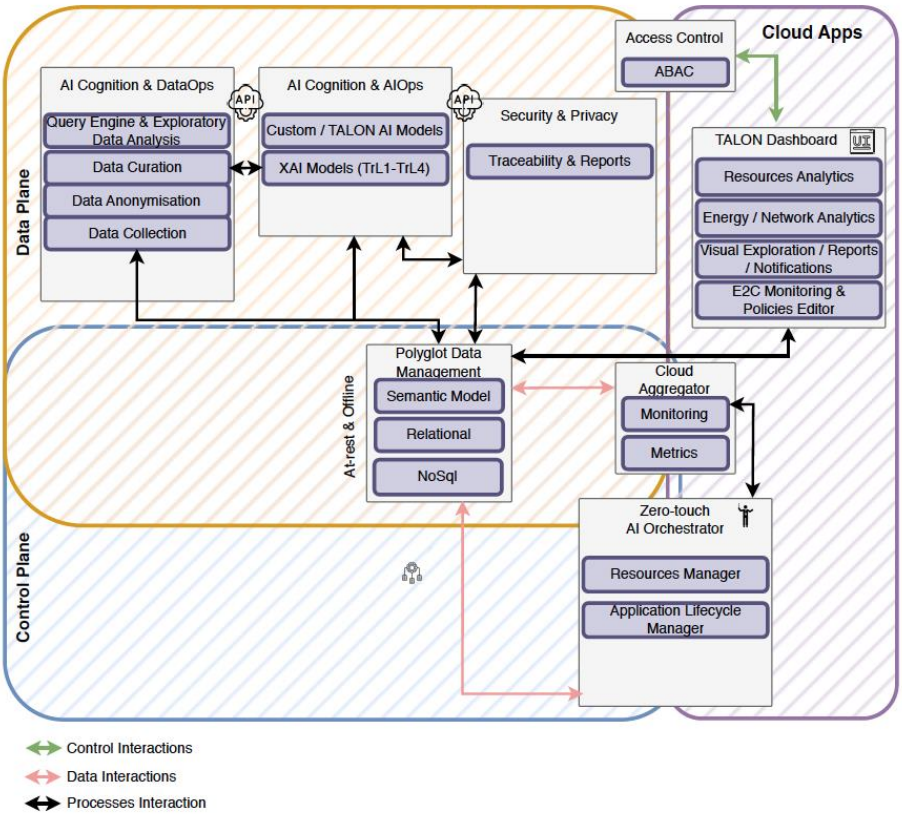

Figure 22: UC4 Architectural Modules to be Tested (CERTH pilot)

3.4.2 Demonstration Planning

The demonstration of the pilot will be done in two stages;

(a) On the outside of CERTH building, in a specific area, where employees will act as workers in a factory wearing or not wearing safety equipment, in order the drone to be able to detect the necessary objects. Multiple participants will be involved in this demonstrating scenario to ensure a fair dataset and analysis of the tools. This will be a trial of the setting offered by CERTH pilot.

(b) The final scenario will be to fly the drone in an industry's indoor environment; possibly at the FACTOR premises. The drone will fly inside the factory to detect dangerous situations/conditions working, e.g., detect workers that work without helmets, glasses, etc.

The development of the tools that will be used in CERTH pilot, are in progress. In the figure below you can see which modules will be ready for demonstration on M22, M30 and M36 by the end of the project:

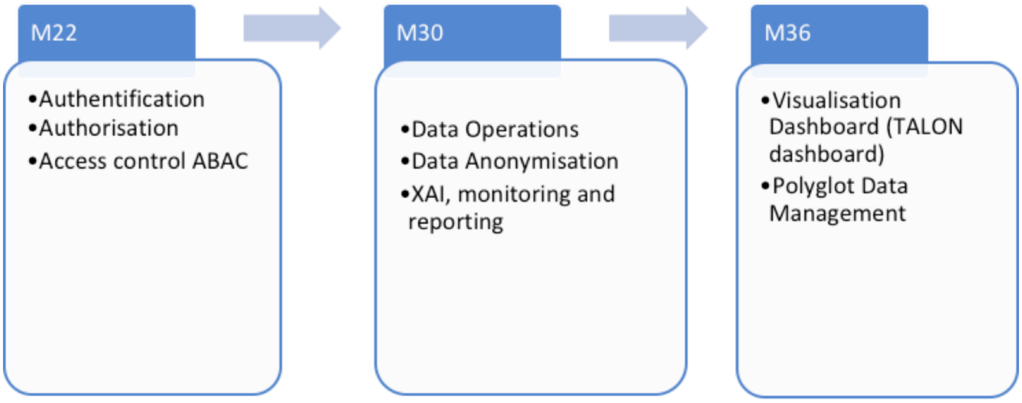

Figure 23: UC4 Demonstration Planning (CERTH pilot)

4 Evaluation Methodology

This chapter is focused on the methodology that will be used to evaluate the overall TALON project. Specifically, here we elaborate on the overall methodology that will be followed to demonstrate how the non-functional TALON requirements are met. This procedure would focus on the V-model SDLC methodology.

4.1 Verification and Validation Model

The V-model, also known as the Verification and Validation model, is a software development lifecycle (SDLC) methodology that emphasizes a systematic approach, ensuring the quality of a software system. It establishes a clear relationship between the development and testing phases, creating a V-shaped diagram as shown in Figure 24 where development activities on the left side correspond to their respective testing activities on the right side. It is based on the association of a testing phase for each corresponding development stage. The development of each step is directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e., for each development activity, there is a testing activity corresponding to it. This structured approach promotes early identification and rectification of defects, leading to a more robust and reliable final product.

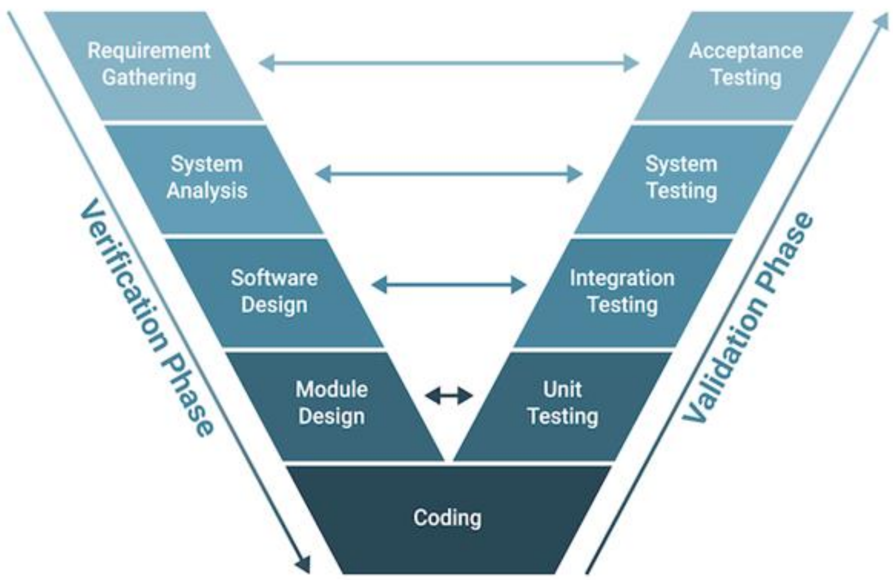

Figure 24: SDLC Verification and Validation Model.

Among other SDLC methodologies, the unique feature of V-model lies in its multiple benefits. Early defect detection is a key strength, achieved by integrating testing activities alongside development. This not only reduces the cost of fixing problems but also prevents them from snowballing into later stages, saving both time and resources. Another strength of this model is that it creates clear communication channels between developers and testers through a well-defined roadmap. This roadmap translates to corresponding testing activities for each development phase, ensuring all stakeholders have a shared understanding of project goals and expectations. Furthermore, the emphasis on thorough testing throughout the lifecycle leads to a higher quality final product. By systematically testing against pre-defined requirements, the V-model helps guarantee the software meets its intended functionality, performance, and user experience goals. Additionally, it offers a clear and organized framework for development, providing greater control over the entire process. This structured approach is particularly valuable for large-scale projects with complex requirements such as Talon. Finally, the V-model helps mitigate project risks by proactively addressing potential issues early on through testing, which reduces the risk of encountering major defects later in the development cycle, leading to a more predictable and reliable development process.

4.1.1 Verification phases

It involves a static analysis technique (review) done without executing code. It is the process of evaluation of the product development phase to find whether specified requirements are met.

There are several Verification phases in the V-Model:

Business Requirement Analysis:

Initial phase involves detailed communication with the customer to understand requirements and expectations, crucial for aligning the project scope with customer needs. Acceptance test design planning is conducted to address uncertainties in customer requirements.

System Design:

Once requirements are clear, the system is designed comprehensively, focusing on architecture and functionality. This phase lays the foundation for future test case execution and ensures alignment with project goals.

Architectural Design:

In this stage, architectural specifications are comprehended and designed. Usually, several technical approaches are put out, and the ultimate choice is made after considering both the technical and financial viability. The system architecture is further divided into modules that each handle a distinct function. At this point, the exchange of data and communication between the internal modules and external systems are well understood and defined. During this phase, integration tests can be created and documented using the information provided.

Module Design:

This phase, known as Low-Level Design (LLD), specifies the comprehensive internal design for every system module. Compatibility between the design and other external systems as well as other modules in the system architecture is crucial. Unit tests are a crucial component of any development process since they assist in identifying and eradicating the majority of mistakes and flaws at an early stage. Based on the internal module designs, these unit tests may now be created.

Coding Phase:

The Coding step involves writing the code for the system modules that were created during the Design phase. The system and architectural requirements are used to determine which programming language is most appropriate. The coding standards and principles are followed when performing the coding. Before the final build is checked into the repository, the code undergoes many code reviews and is optimized for optimal performance.

4.1.2 Validation phases

This section is involved with the dynamic analysis techniques (functional, and non-functional), and testing done by executing code. Validation is the process of evaluating the software after the completion of the development phase to determine whether the software meets the customer's expectations and requirements.

As shown in Figure 24, there are several validation phases:

Unit Testing:

Unit Test Plans are developed during the module design phase. These Unit Test Plans are executed to eliminate bugs in code or unit level.

Integration testing:

After completion of unit testing Integration testing is performed where the different modules are integrated to the system, which afterwards is being tested. Integration testing is performed in the Architecture design phase. This test verifies the communication of modules among themselves.

System Testing:

System testing tests the complete application with its functionality, inter-dependency, and communication. It tests the functional and non-functional requirements of the developed application.

User Acceptance Testing (UAT): UAT is performed in a user environment that resembles the production environment. UAT verifies that the delivered system meets the user's requirement, and the system is ready for use in the real world.

4.2 Mapping non-functional requirements to the V-model

Following the previous section regarding the V-model architecture, this section focuses on the mapping of Talon non-functional requirements (NFRs) to the Verification and Validation methodology. Employing the V-Model methodology ensures a comprehensive approach to validating and verifying these NFRs throughout the project lifecycle. This document outlines the application of the V-Model framework to address each NFR defined within the project scope.

In terms of compatibility, the TALON system must demonstrate seamless coexistence with other software and effective interoperability with external systems. To achieve this, verification activities include system design reviews to ensure architecture isolates components and utilizes well-defined communication protocols. Unit testing further validates proper interaction without interference and successful data exchange with external systems.

Functional suitability is crucial for the TALON system, necessitating both completeness and correctness of its functionalities. Requirements analysis ensures clear documentation of all functionalities, with a traceability matrix linking each requirement to system components. System testing with comprehensive test cases validates the delivery of all intended functionalities, while unit testing ensures individual components perform correctly according to specifications.

Maintainability is addressed through modifiability, modularity, reusability, and testability. Design reviews focus on architecture analysis to promote modularity and clear interfaces, facilitating ease of modification without disrupting overall system behavior. Real-world deployment and usability testing validate adaptability to different platforms and environments and assess user accessibility and intuitiveness.

Performance considerations for the TALON system include capacity, resource utilization, and time behavior. Architecture analysis during system design reviews identifies resource-efficient technologies, while unit testing monitors resource usage under various loads. System testing with increased load measures actual resource utilization and ensures efficient capacity and acceptable latency and throughput.

Reliability is ensured through availability, fault tolerance, and recoverability mechanisms. Redundancy mechanisms and failover strategies are assessed during system design reviews, with unit testing simulating component failures and validating fault tolerance mechanisms. System testing under failure scenarios ensures high availability and successful recovery without data loss.

Security features of the system encompass accountability, authenticity, confidentiality, and integrity. Verification activities include system design reviews and unit testing of security mechanisms, with validation involving testing under various security scenarios to ensure robustness.

In terms of usability, the TALON system must be accessible and intuitive for users. Design reviews assess user interface accessibility and intuitiveness, with unit testing involving testing with assistive technologies. Usability testing with representative users further validates accessibility and appropriateness recognizability.

5 KPIs Definition

This section reports the progress that has so far being made on TALON's Key Performance Indicators. This deliverable receives its input from WP2, specifically D2.1 "*Use Case, KPIs, Requirements, Specification, Slices & Technology Enablers Definition Report*", since WP5 is associated with the overall project evaluation. The initial KPIs definition was reported on D2.1 to establish a baseline for assessing the project's progress and performance. During TALON's technical review, it was determined that these initial KPIs will be further refined to better align with the project's evolving objectives and insights gained from early implementation stages. The KPIs will be redefined, along with their corresponding descriptions, and will be presented in the revision of D2.1 at M24 (Sept 2024) and at D5.4 "*Final TALON Platform Setup, Operation, Continuous Integration & Maintenance Report*" end of the project (M36) having completed all the validation activities.

6 Conclusion and Future Outlook

The deliverable D5.1 “Installation & Demonstration Planning, Evaluation Methodology & KPIs Definition Report” reports the work carried out in the context of T5.1 “Installation & Demonstration Planning, Evaluation Methodology & KPIs Definition”. The document provides:

- Information about the implementation status and deployment planning of each architectural module included in the TALON’s architecture as it was presented in D3.1 "Architecture & Platform Design Blueprint" report.
- A detailed demonstration planning aligning the technical with the demonstration activities and providing a detailed description of the M22, M30 and M36 demonstration targets.
- The evaluation methodology followed by the TALON project in order to ensure a comprehensive evaluation.
- A status report on the KPIs definition progress.

D5.1 is the first out of four deliverables related to the “Integration, Validation & Demonstration” work package of TALON, which will be followed by D5.2 “Initial TALON Platform Setup, Operation, Continuous Integration & Maintenance Report”. The latter deliverable will leverage the components described in the present document towards providing and releasing a first integrated installation of the TALON platform that will support the deployment and operation of the demonstrators.

References

[1] A. P. Jönsson, "Named-entity recognition with BERT for anonymization of medical records," https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1566701&dswid=8313.

[2] Kian Long Tan, Chin Poo Lee, and Kian Ming Lim, "RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis," https://www.mdpi.com/2076-3417/13/6/3915, 2023.

[3] Constantinos Patsakis and Nikolaos Lykousas, "Man vs the machine in the struggle for effective text anonymisation in the age of large language models," https://www.nature.com/articles/s41598-023-42977-3, 2023.

[4] Damian P. Kotevski, Robert I. Smee, Matthew Field, Yvonne N. Nemes, Kathryn Broadley, and Claire M. Vajdic, "Evaluation of an automated Presidio anonymisation model for unstructured radiation oncology electronic medical records in an Australian setting," https://www.mdpi.com/2076-3417/13/6/3915, 2022.

Funded by
the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070181