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

Deliverable

D3.1 Architecture & Platform Design Blueprint

Actual submission date: 01/10/2023

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
D3.1 Architecture & Platform Design Blueprint			
Work Package:	WP3		
Lead partner:	ENG		
Author(s):	Flavio Scaccia (ENG);		
Reviewers:	FACT, UL		
Due date:	29/09/2023		
Deliverable Type:	R - Document, report	**Dissemination Level:**	PU - Public
Version number:	1.0		

Revision History

Version	Date	Author	Description
0.1	17/07/2023	Flavio Scaccia (ENG)	First release of the ToC
0.2	07/09/2023	Sophia Karagiorgou, Ioannis Pastellas, Petros Petrou (UBI) Salvatore Cipolla, Flavio Scaccia (ENG) Konstantinos Xouveroudis, Stathis Mavridopoulos, Vasileios Gavresis, Nikolaos Moschos, Ioannis Bothos, Dimitrios Asimopoulos, Nikolaos Episkopos. (MINDS) Konstantinos Kyranou, Ioannis Hadjigeorgiou (SID) George Delaportas (PROBO)	Input provided by partners

		Konstantinos Ntontin (UL) Stylianos Trevlakis (IC) Simon Sarkissian, Lydia Markopoulioti Titos Georgoulakis (8BELLS) Faustino Alarcón Valero, Foivos Psarommatis (UPV) Georgia Apostolou (CERTH) Thomas Lagkas (IHU) Marcello Morchio (TEI)	
0.3	08/09/2023	Flavio Scaccia (ENG)	Sent for internal quality and content review
0.4	15/09/2023	Konstantinos Ntontin (UL), Alfredo Giménez (FACT)	Content review
0.5	27/09/2023	Flavio Scaccia (ENG), Sophia Karagiorgou (UBI), Dimitris Asimopoulos (MINDS), Gino Ciccone (TEI)	Addressed reviewers' comments and revisions
0.6	29/09/2023	Flavio Scaccia (ENG)	Final version, addressed quality assurance check comments
1.0	01/10/2023	Susanna Bonura (ENG)	Final coordinator review before submission

Table of Contents

List of Figures 6

List of Tables 7

Definitions and acronyms 8

1 Introduction 11

1.1 Objectives of the Deliverable 12

1.2 Relation with other Tasks and Deliverables 12

1.3 Document Structure 12

2 Architecture Componentisation 14

2.1 TALON High-Level Architecture and Defined Components 14

2.2 Architecture Walkthrough & TALON Actors 17

3 Access and Security 21

3.1 Authentication and Authorisation 21

3.1.1 Business Logic 21

3.1.2 Functional Description 22

3.1.3 Information Flow and Internal Architecture 23

3.1.4 Interaction with Other Components 23

3.2 Data Anonymisation 24

3.2.1 Business Logic 24

3.2.2 Functional Description 24

3.2.3 Information Flow and Internal Architecture 25

3.2.4 Interaction with Other Components 27

3.3 DLTs for Securing AI/ML models weights 27

3.3.1 Business Logic 27

3.3.2 Functional Description 28

3.3.3 Information Flow and Internal Architecture 28

3.3.4 Interaction with Other Components 29

3.4 Anomaly Detection 30

3.4.1 Business Logic 30

3.4.2 Functional Description 30

3.4.3 Information Flow and Internal Architecture 31

3.4.4 Interaction with Other Components 31

4 AI-fuelled Orchestration 33

4.1 Application Lifecycle Manager 33

4.1.1 Business Logic 33

4.1.2 Functional Description 34

4.1.3 Information Flow and Internal Architecture 34

4.1.4 Interaction with Other Components 35

4.2 NG-SDN and Distributed Intelligence 35

4.2.1 Business Logic 35
4.2.2 Functional Description 36
4.2.3 Information Flow and Internal Architecture 36
4.2.4 Interaction with Other Components 37
4.3 AI Swarm Orchestration 37
4.3.1 Business Logic 37
4.3.2 Functional Description 38
4.3.3 Information Flow and Internal Architecture 38
4.3.4 Interaction with Other Components 39
4.4 Resource Allocation and Deployment 40
4.4.1 Business Logic 40
4.4.2 Functional Description 40
4.4.3 Information Flow and Internal Architecture 40
4.4.4 Interaction with Other Components 41
4.5 Smart Policy Manager 42
4.5.1 Business Logic 42
4.5.2 Functional Description 42
4.5.3 Information Flow and Internal Architecture 43
4.5.4 Interaction with Other Components 43
4.6 Data Monitoring, Collection and Aggregation 44
4.6.1 Business Logic 44
4.6.2 Functional Description 44
4.6.3 Information Flow and Internal Architecture 45
4.6.4 Interaction with Other Components 46
4.7 AI Model Training, Adaptations and SLOs 47
4.7.1 Business Logic 47
4.7.2 Functional Description 47
4.7.3 Information Flow and Internal Architecture 48
4.7.4 Interaction with Other Components 48
5 AI Cognition 50
5.1 Self-healing and Self-correcting 50
5.1.1 Business Logic 50
5.1.2 Functional Description 50
5.1.3 Information Flow and Internal Architecture 50
5.1.4 Interaction with Other Components 51
5.2 AI Capabilities, Optimised and Transfer Learning 51
5.2.1 Business Logic 51
5.2.2 Functional Description 52
5.2.3 Information Flow and Internal Architecture 53
5.2.4 Interaction with Other Components 53

5.3 Data Operations *54*
5.3.1 Business Logic *54*
5.3.2 Functional Description *54*
5.3.3 Information Flow and Internal Architecture *55*
5.3.4 Interaction with Other Components *56*
5.4 Digital Twin *57*
5.4.1 Business Logic *57*
5.4.2 Functional Description *58*
5.4.3 Information Flow and Internal Architecture *58*
5.4.4 Interaction with Other Components *59*
5.5 XAI and Monitoring *59*
5.5.1 Business Logic *59*
5.5.2 Functional Description *60*
5.5.3 Information Flow and Internal Architecture *61*
5.5.4 Interaction with Other Components *61*
5.6 Polyglot Data Management *62*
5.6.1 Business Logic *62*
5.6.2 Functional Description *63*
5.6.3 Information Flow and Internal Architecture *64*
5.6.4 Interaction with Other Components *65*
5.7 Visualisation Dashboard *65*
5.7.1 Business Logic *65*
5.7.2 Functional Description *65*
5.7.3 Information Flow and Internal Architecture *66*
5.7.4 Interaction with Other Components *66*
6 Requirements Traceability Matrix 67
7 Partners assets and background technologies 71
8 Conclusions 73
References 74

List of Figures

Figure 1 - TALON Conceptual Architecture 17
Figure 2 - TALON's UML Diagram, User Roles and Interaction with Processes 20
Figure 3 - Authentication and Authorisation component architecture 23
Figure 4 - Numerical and Text Anonymisation architecture 26
Figure 5 - Face Anonymisation architecture 26
Figure 6 - Distributed Ledger Technology component architecture 29
Figure 7 - Anomaly Detection component architecture 31
Figure 8 - Application Lifecycle Manager Component architecture 35
Figure 9 - NG-SDN and Distributed Intelligence Component architecture 37
Figure 10 - Swarm Orchestration sequence diagram 39
Figure 11 - Resource Manager architecture 41
Figure 12 - Smart Pricing architecture 43
Figure 13 - Data Monitoring, Collection and Aggregation sequence diagram 45
Figure 14 - Data Monitoring, Collection and Aggregation architecture 46
Figure 15 - AI Model Training, Adaptations and SLOs Component architecture 48
Figure 16 - Self-healing and Self-correcting component architecture. 51
Figure 17 - AI Capabilities and Transfer Learning architecture 53
Figure 18 - Information flow of the TALON data preprocessing component and interaction with other components 55
Figure 19 – Data curation architecture 56
Figure 20 - Digital Twin workflow diagram. 59
Figure 21 - XAI architecture 61
Figure 22 - Polyglot Data Management component architecture 65

List of Tables

Table 1 - User Role Categories 18
Table 2 - Stakeholder Categories 18
Table 3 - Authentication and Authorisation functional description 22
Table 4 - Authentication and Authorisation interactions 23
Table 5 - Data Anonymisation functional description 24
Table 6 - Data Anonymisation interactions 27
Table 7 - DLT functional description 28
Table 8 - DLT interactions 29
Table 9 - Anomaly Detection functional description 30
Table 10 - Anomaly Detection interactions 32
Table 11 - Application Lifecycle Manager functional description 34
Table 12 - Application Lifecycle Manager interactions 35
Table 13 - NG-SDN and Distributed Intelligence functional description 36
Table 14 - NG-SDN and Distributed Intelligence interactions 37
Table 15 - Swarm Orchestration functional description 38
Table 16 - Swarm Orchestration interactions 39
Table 17 - Resource Manager functional description 40
Table 18 - Resource Manager interactions 42
Table 19 - Smart Policy Manager functional description 42
Table 20 - Smart Policy Manager interactions 44
Table 21 - Data Monitoring, Collection and Aggregation functional description 44
Table 22 - Data Monitoring, Collection and Aggregation interactions 47
Table 23 - AI Model Training functional description 47
Table 24 - AI Model Training interactions 49
Table 25 - Self-healing and Self-correcting functional description 50
Table 26 - Self-healing and Self-correcting interactions 51
Table 27 - AI Capabilities and TL functional description 52
Table 28 - AI Capabilities and TL interactions 53
Table 29 - Data Preprocessing functional description 54
Table 30 - Data Preprocessing interactions 57
Table 31 - Digital Twin functional description 58
Table 32 - Digital Twin interactions 59
Table 33 - XAI and Monitoring functional description. 60
Table 34 - XAI and Monitoring interactions 61
Table 35 - Polyglot Data Management functional description 63
Table 36 - Polyglot Data Management interactions 65
Table 37 - Visualisation Dashboard functional description 65
Table 38 - Visualisation Dashboard interactions 66
Table 39 - Requirements Traceability Matrix 67
Table 40 - Partner Assets 71

Definitions and acronyms

ACS	*Automatic Control System*
AD	*Anomaly Detection*
AI	*Artificial Intelligence*
AR	*Augmented Reality*
BPF	*Berkeley Packet Filter*
CA	*Certificate Authority*
CD/CA	*Collision Detection & Collision Avoidance*
CPU	*Central Processing Unit*
DAG	*Directed Acyclic Graph*
DL	*Deep Learning*
DLT	*Distributed Ledger Technologies*
DNN	*Deep Neural Network*
E2C	*Edge to Cloud*
EU	*European Union*
FL	*Federated Learning*
FR	*Functional Requirement*
GA	*Grant Agreement*
GDPR	*General Data Protection Rights*
GPU	*Graphical Processing Unit*
I4.0	*Industry 4.0*
I5.0	*Industry 5.0*
IEA	*International Energy Agency*
IoT	*Internet of Things*
IT	*Information Technology*
KPI	*Key Performance Indicator*
MCC	*Mission Control Center*
MILP	*Mixed-integer linear programming*
ML	*Machine Learning*
NG-SDN	*Next Generation Software Defined Network*
NLP	*Natural Language Processing*
PII	*Personally Identifying Information*
PKI	*Public Key Infrastructure*
PPE	*Personal Protective Equipment*
QoS	*Quality of Service*
RBAC	*Role-Based Access Control*
RTM	*Requirements Traceability Matrix*
SCs	*Security Configurations*
SLOs	*Service Level Objectives*
SSO	*Single Sign On*
StRs	*Stakeholder Requirements*
TL	*Transfer Learning*
TrL	*Trust Level*
UAVs	*Unmanned Aerial Vehicles*
UCs	*Use Cases*
UUA	*User Authentication and Authorization*
VR	*Virtual Reality*
XAI	*Explainable AI*
YAML	*YAML Ain't Markup Language*
ZDM	*Zero Defect Manufacturing*

Disclaimer

This document has been produced in the context of TALON Project. The TALON project is part of the European Community's Horizon Europe Program for research and development and is as such funded by the European Commission. All information in this document is provided ‘as is’ and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability with respect to this document, which is merely representing the authors’ view.

Executive Summary

The "Architecture & Platform Design Blueprint" details the core concept of TALON, an industry-defining project set to transform Industry 4.0/5.0. This deliverable highlights our commitment to innovative technologies, orchestration, and sustainability.

The document explores the conceptual architecture behind TALON's objectives, focusing on two pivotal planes: the Data Plane and the Control Plane. Our architecture coordinates compute, communication, AI models and data for improved performance and resource efficiency. This work aims to utilise AI for improving the energy efficiency of networks, addressing the crucial requirement for environmentally friendly AI networks as our planet experiences rapid climate change.

An essential element of our architectural design is the incorporation of the stakeholder requirements from previous assignments, as demonstrated by the Requirements Traceability Matrix. Every component described here sets the stage for the realization of a fully integrated and impactful TALON ecosystem.

1 Introduction

The "Architecture & Platform Design Blueprint" document represents the essence of Task 3.1 "Overall Architecture & Platform Design", a critical activity within the TALON project. Task 3.1 aims to deliver a cutting-edge architecture that can handle a variety of Industry 4.0 and Industry 5.0 use cases and challenges while transforming the industrial environment.

The introduction of AI infrastructure in the industry can assist manufacturers in overcoming several obstacles and enhancing their operations. Here are the significant issues that the introduction of AI infrastructure can resolve:

- **Inflexibility**: Traditional manufacturing systems are often inflexible and cannot adapt to changing demand. This means that these systems must be able to adapt to changing demand and requirements without disrupting production. AI infrastructure can help to make manufacturing systems more flexible and responsive to change.
- **Complexity**: Manufacturing systems are becoming increasingly complex, due to the use of automation and data exchange. AI infrastructure can help to manage the complexity of manufacturing systems and make them easier to operate and maintain.
- **Cost**: The cost of manufacturing is increasing, due to factors such as rising energy prices and the need to comply with regulations. AI infrastructure can help to reduce the cost of manufacturing by improving efficiency and productivity.
- **Sustainability**: Manufacturing is a major contributor to environmental pollution. AI infrastructure can help to make manufacturing more sustainable by reducing energy consumption and waste.

At the centre of this architecture lies the pursuit of dynamic scalability, achieved by orchestrating computational and communication resources, AI models, and data, while prioritizing energy efficiency. The mission to develop greener AI networks is driven by the necessity to reduce energy consumption considering the ongoing global climate crisis, making a significant contribution to sustainability efforts. Indeed, recent reports from authoritative sources such as the International Energy Agency (IEA), which tracks and reports on global energy use and emissions, show that the global CO2 balance is split between electricity and heat production, followed by manufacturing, and then transport and buildings. Manufacturing takes second place in this ranking and accounts for around 20% of global carbon emissions[1], a figure that can change depending on a number of factors, including the energy mix of different countries and regions and the level of industrial activity.

In addition to dynamic scalability, AI infrastructure can also be used to improve other aspects of manufacturing, such as:

- Quality control: AI can be used to identify defects in products or processes.
- Predictive maintenance: AI can be used to predict when equipment will fail, so that maintenance can be scheduled before a breakdown occurs.
- Resource optimization: AI can be used to optimize the use of resources, such as energy and materials.

The rapid evolution of technology, which is impacting Industry 4.0, is paving the way for the emerging concept of Industry 5.0, which is on the horizon. This concept represents the next phase in the

[1] *CO2 Emissions in 2022 - https://www.iea.org/reports/co2-emissions-in-2022*

evolution of industrial processes, with a strong emphasis on human-machine synergy. Companies are likely to rely more on self-learning systems to increase automation and efficiency, and advanced robotic systems and artificial intelligence will enhance this collaboration. For example, robots can take over physically demanding or dangerous tasks, while humans use their expertise for complex decision-making and monitoring tasks. Industry 5.0 offers companies huge opportunities to automate processes, reduce costs and improve product quality. Using advanced sensors and connected systems, production processes can be fully optimised to achieve greater efficiency and productivity, improve working conditions in real time and avoid critical situations. In addition, the combination of artificial intelligence and machine learning methods helps to identify problems early and take countermeasures.

The industry challenges described are addressed in TALON, and this document represents its architecture as a basis for development. The meticulous study of the building's components is the focus, which provides an exhaustive and comprehensive description of their interfaces and internal logic. The design considers essential aspects, including access and security methods, to ensure strong protection of sensitive data and protection against potential threats. It also explores the details of the AI orchestrator, which coordinates the intelligent allocation of resources to enhance system effectiveness and performance. The document then provides a light on the field of AI cognition, revealing the system's potential for self-healing, self-correcting, and advanced learning capabilities.

To ensure alignment with the needs and aspirations of stakeholders, the activity connected to the construction of the architecture also includes the integration of prerequisites obtained from previous tasks (see D2.1 for further details). A traceability matrix serves as a better description of this integration. The document also emphasizes the partner assets used in this architectural project, highlighting the partners' participation in a collaborative effort and utilizing their combined knowledge. The created architecture drives TALON toward its mission for changing the Industry 4.0/5.0 environment.

1.1 Objectives of the Deliverable

The objective of D3.1 "Architecture & Platform Design Blueprint" is to identify the fundamental architectural components and their interfaces. This will lead to the creation of an architecture that supports various aspects such as high energy efficiency and dynamic scalability. The architecture will facilitate the synchronisation of infrastructures, AI models and data with the various resources and communication systems behind them. It will also define an approach for distributing IT services across the E2C continuum by making them migratable between different layers according to the NG-SDN paradigm.

1.2 Relation with other Tasks and Deliverables

Task 3.1 uses valuable insights from D2.1 to design the architecture starting from the functional requirements, ensuring it aligns with the use cases' objectives and success metrics: Deliverable D2.1 is a crucial input for D3.1, because it identifies supported Use Cases (UCs) and determines the platform's requirements and stakeholders, laying the foundation for the project's technical stages and helps partners understand the UC scenarios.

Additionally, this deliverable will contribute to D3.6, which focuses on validating and demonstrating TALON's overall architecture. This process involves establishing communication among components and transforming the architecture into a functional and purposeful system.

1.3 Document Structure

The structure of the rest of this document is as follows:

Chapter 2 outlines the identified TALON actors and gives a thorough explanation of the high-level components that comprise the architecture.

The fundamental layer of the architecture is represented by Chapters 3, 4, and 5. These chapters individually provide a comprehensive explanation of the functions and interfaces of the corresponding components within their respective domains.

Chapter 3 specifically focuses on access and security with the goal of implementing a range of security measures that uphold the confidentiality, integrity, and availability of both data and systems.

Chapter 4 delves into the process of coordinating dynamic interactions within the TALON framework, referred to as AI orchestration.

Finally, Chapter 5 explores AI cognition and employs a variety of generic and domain-specific analysis algorithms, digital twins, self-healing capabilities, XAI methods, and data processing techniques.

Chapter 6 shows the integration of stakeholder requirements into the functional requirements covered by each component. The traceability matrix makes sure the needs of stakeholders are addressed.

Chapter 7 provides a comprehensive list of the assets and contributions made by partners participating in the TALON project.

2 Architecture Componentisation

2.1 TALON High-Level Architecture and Defined Components

The goals of the TALON System are to:

- Enable zero-touch deployment and operation
- Reduce the energy footprint of the whole AI network
- Guarantee high-level in security and privacy
- Assess and boost the AI Edge-to-cloud performance
- Enable reusability of datasets, algorithms, models
- Present an AI theoretical framework
- Boost AI explainability and transparency

These goals will be delivered by an architecture that lies in the pursuit of dynamic scalability, achieved by orchestrating computational and communication resources, AI models, and data, while prioritizing energy efficiency. This architecture will be materialized through two planes, i.e., the **Data Plane** and the **Control Plane** as depicted on Figure 1. As depicted each plane comprises a set of discrete components that interact with each other.

The **Data Plane** is responsible for the efficient data lifecycle management of the TALON Layers and components, i.e., data generated and used by the TALON applications and monitoring data collected to provide the AI-fuelled Orchestration services. The **Control Plane** is responsible for the synchronisation of the TALON's internal processes, the alerts and the objectives (i.e., metadata and metrics collected for different constraints) which are translated and enforced by means of policies (e.g., resource, operational, deployment, etc.). The Conceptual Architecture of TALON has three (3) main utility Layers which are the: (i) Access and Security Layer, (ii) AI-fuelled Orchestration Layer, and (iii) AI Cognition Layer. The Access and Security Layer brings the functionalities of securing and preserving the privacy of TALON's services, user's personal data and reporting system's abnormal events. The AI-fuelled Orchestration Layer is responsible to aggregate monitoring data from edge and cloud applications, interact with the AI Cognition Layer which brings the AI Models and enforce runtime adaptations and smart decisions. Last, the AI Cognition Layer provides all the data management and enhancement operations, as well as the training of the different AI Models to both support the end user applications, modelling and predictions of the AI-fuelled Orchestration Layer. The components belonging to each Layer and the functionalities each Layer supports are detailed in the following paragraphs.

In the TALON Conceptual Architecture, an **Access and Security Layer** supports set of security and privacy-preserving services that will be embedded in the Use Cases and addresses the diverse security specifications of all the computing resources and data sources. These are mainly revolving around: (i) network and data access services through the provision of advanced authentication protocols and secure communication channels; (ii) data anonymisation methods for privacy at the user level; (iii) Distributed Ledger Technologies (DLTs) for securing Federated Machine Learning (FML) models characteristics (e.g., weights, timestamps, etc.), and (iii) anomaly detection for the prompt identification and reporting of behaviourally suspicious traffic patterns.

Regarding the **AI-fuelled Orchestration Layer**, several components interact within the TALON ecosystem. Existing cloud-orchestrators (such as Docker Swarm, Kubernetes) maintain their own

version of Directed Acyclic Graph (DAG) representation (e.g., docker-compose[2], Helm charts[3]). TALON will maintain backward compatibility with the de-facto DAG models. However, on top of these models, specific modelling extensions will be provided that will drive sophisticated orchestration logic. These metadata include, mainly, deployment constraints, resource constraints, operational constraints and security constraints.

For example, let us assume an application that aims to capture live video from several drones and use these streams in order to perform both local and remote data processing. Such an edge application may require specific edge nodes with minimum memory of 1GB to host an application specially built using ARM extensions. The application must be deployed and monitored from a specific geography region and entails specific QoS characteristics since it will generate process and transmit a video stream. These characteristics refer to the acceptable maximum-jitter, maximum-delay, minimum-throughput and maximum-packet drop that is acceptable for a given threshold of QoS.

The streamed data have additional requirements from the “backhaul” part of the network in terms of specific storage and compute power that is required in order to facilitate transcoding business logic. Such business logic may require special virtualization extensions from the data-centre hypervisors. During the operation of the entire application graph, the service provider decides to push specific images to the edge nodes for them to perform on-the-fly analysis pattern recognition of these images. The provider cares about the integrity and the confidentiality of these images. It is of high importance the running application remains intact and immune to a wide variety of privacy and/or security issues. As such, additional security constraints regarding the user privacy (e.g., anonymised imagery, numerical and textual data) and security (e.g., attributed-based access control) will be managed by the respective components. All these constraints will be managed by the Application Lifecycle Manager.

Upon initial deployment, the Application Lifecycle Manager will undertake the tasks of maintaining a valid scalable and secure application graph. The operation of this component can be modelled as an automation where a set of desired states are dynamically extracted upon evaluation of the as-is state. The component is trying to maintain an equilibrium of the “desired” versus the “as-is” state taking into consideration several external (to-the-loop) factors such as runtime policies and alternations of the offered resources. The Application Lifecycle Manager consists of several internal modules that undertake distinct tasks such the interaction with the Cloud and Edge Aggregator (i.e., metrics about the physical resources managed by the Data Monitoring, Collection and Aggregation), the interaction with the Analytics Engine supported by the AI Models Training, Adaptations and Service Level Objectives (SLOs), the identification of the best-of-breed action that should be enacted in the form of a policy (managed by the Smart Policy Manager).

In addition to the Application Lifecycle Manager, the centralized orchestration backend is complemented by the Resources Manager, the Cloud Aggregator, and the Analytics Engine components. The Resources Manager has a two-fold goal. On the one hand it will act as an abstraction layer to both “stationary” cloud resources and “dynamic” edge resources. Edge resources will be reported by an “appointed” cluster-head that represents a regional deployment.

On the other hand, the Cloud and Edge Aggregator will be responsible to populate metrics that relate to both the consumed resources and the activated policies (i.e., application-specific metrics). A key research issue in this field is how to restrict the bandwidth and the respective throughput of the monitoring streams. A sophisticated hierarchical structure will be adopted through the programmable

[2] https://docs.docker.com/compose/
[3] https://helm.sh/docs/topics/charts/

Control Plane realised by centralised Software-Defined Networking (SDN) mechanisms. Distributed Intelligence will be ensured by also collecting metrics at network level to compute adaptable policies and corrective mechanisms for the networking of E2C services. The policies will be directed to the centralised SDN mechanisms through a novel programmable and controllable paradigm. This paradigm enables to enforce high-level actions by guiding underlying switches to handle data flows throughout the communication network mesh of E2C services. Given the global visibility, the SDN mechanisms will be capable of optimising network-wide data flows more efficiently.

Finally, when a service graph is deployed TALON will allow the execution of analytic queries for datasets that are generated/consumed within the lifecycle of the application. To do so, a set of complementary components will be developed that map to all layers mentioned above. More specifically, an Analytics Engine will undertake the task of query-planning and coordination (at the orchestration layer) and a Polyglot Data Management component will undertake the task of data persistency, both in terms of sending/receiving, storing and retrieving data from the TALON services as well as various aspects of information that is being generated and consumed within the lifecycle of the various components complimented by a message queueing system (i.e., the Message Broker). It is also responsible to execute the sub-queries according to the coordinated plan. Such queries may (indicatively) refer to specific data aggregations on application-specific data (e.g., aggregate-function-X on IoT sensor readings for time-range-Y) or on operational/performance data (e.g., number of transcoded frames per second for deployment X). The Polyglot Data Management component supports all the data modalities for integrating heterogeneous data with different attributes coming from the TALON Use Cases.

Last, as far as the **AI Cognition Layer** is concerned, it will deliver various Artificial Intelligence (AI), explainable AI (XAI) and Data Operation methods by means of AIOps and DataOps through the different TALON components. A set of Generic & Domain Specific Analytic Algorithms (i.e., digital twins, self-healing and self-correcting component, XAI methods for data trustworthiness, hybrid and optimised learning, AI capabilities and transfer learning) will be developed and interconnected among each other in terms of input/output data streams/objects. These algorithms will be delivered by means of AIOps. Upon their execution, the outcome will be exposed to the Visualisation Dashboard. The Visualisation Dashboard assembles in a unified user interface all the envisioned components, which enable stakeholders to gain insights from the reports, analytics and alerts produced per component. The end user data lifecycle is being managed by concrete Data Operations and workflows by means of DataOps. Data are ingested through the Data Collection service and are stored in the Polyglot Data Management. Prior to their storage, data pass through the Data Curation in order to ensure the veracity, timeliness, and transparency. The Data Curation performs a set of quality checks to discover inconsistencies, missing values and other anomalies in the data and eventually ensures their integrity and completeness, to be followed by several associated with data cleaning procedures (i.e., data records removal, imputation, replacement based on the rule enforced). Then, data are semantically annotated and harmonized through the Data Enrichment service to promote data interoperability and reuse. Data-oriented enrichment helps to develop robust and flexible annotations and provide a valuable source for common representation of similar concepts for disambiguation purposes. The curated and semantically enriched data are stored in the Polyglot Data Management. Both the data and annotations are made available to the Query Engine & Exploratory Data Analysis for further exploration, analysis, assisted query building and visualisation. The Query Engine & Exploratory Data Analysis provides to the end users an intuitive environment to select the preferable datasets, define and execute queries on the available data of the Polyglot Data Management.

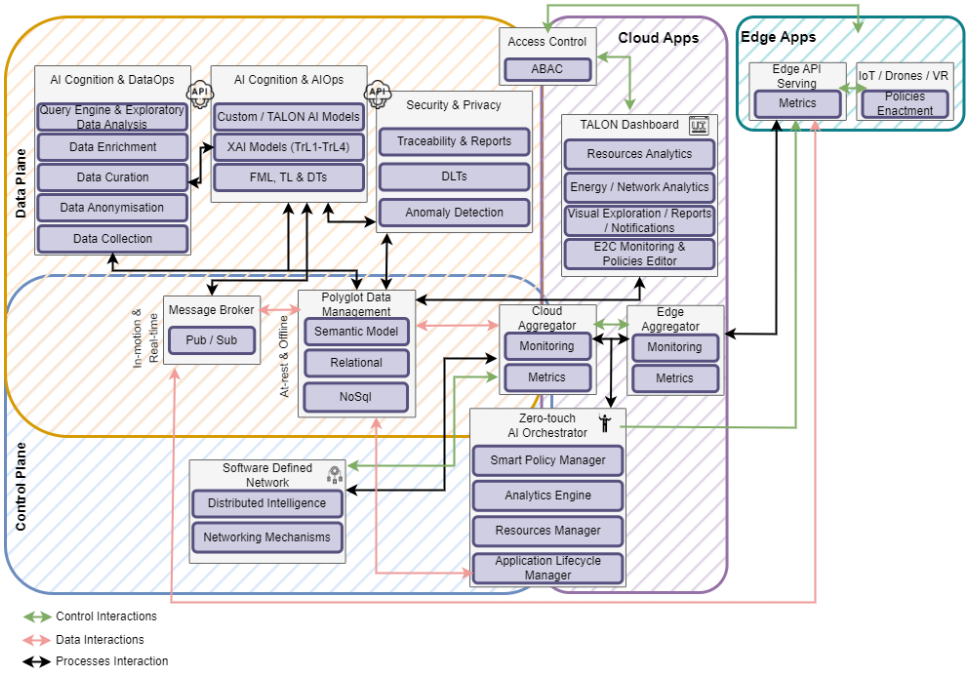

Figure 1 - TALON Conceptual Architecture

System and software engineering plays a pivotal role in assessing the developed solution's functionality, reliability, and security. In D2.1-Use Case, KPIs, Requirements, Specification, Slices & Technology Enablers Definition Report, we adopted the *ISO/IEC Standard 25010:2011*[4] to define the qualities, or overall characteristics of the TALON System and we reported its relationship with the TALON Requirements. In this deliverable, we derive a Requirements Traceability Matrix (RTM) to continuously monitor the evolution of the technical progress and its alignment with the user requirements. To derive the conceptual view of the TALON architecture, we adopted key contents from the *ISO/IEC/IEEE 42010:2011*[i5]. This standard addresses the creation, analysis and sustainment of architectures of systems through the use of architecture descriptions. The outcome of these descriptions is the establishment of the conceptual model of the architecture. description is established. The required contents of an architecture description are specified. In the TALON Conceptual Architecture, we adopted different architecture viewpoints, and architecture frameworks detailed on a per component basis in the following sections.

2.2 Architecture Walkthrough & TALON Actors

The multiple benefits that E2C computing offers are broadening the range of potential user roles and stakeholders who may interact with the TALON System. These diverse user roles can derive advantages from this technology, whether they are individual users or organisations, from three different positions: (i) use services deployed at the edge and/or cloud, (ii) develop services deployed at the edge and/or cloud, and (iii) design and operate edge infrastructures. These positions form the three categories of user roles with the TALON System are presented in Table 1.

[4] https://www.iso.org/standard/35733.html
[5] https://www.iso.org/standard/50508.html

Table 1 - User Role Categories

Categories	Description
End user of a service / application	They could be individuals or groups of users who are interested in using edge technology to enhance their approaches, to solve a problem in their market, to outperform competition etc.
Service developers / providers	They could be individuals or companies that possess the technical knowledge to develop and support Edge to Cloud (E2C) and AI applications. The main motivation for this layer of user roles, is to expand their expertise and operations to new markets using state of the art tools and technologies, having revenue as their final goal.
E2C infrastructure providers	This category of stakeholders includes large companies operating in the Information Technology market, who produce devices throughout the (IT) ecosystem (e.g., network devices, IoT devices, standalone micro-processors etc.) or who provide their infrastructure for communication establishment (e.g., data centres, etc.).

The TALON System is complex and potentially engages multiple stakeholder roles to fully exploit all its components and make it altogether available as a turn-key solution. Based on the technical analysis of the platform and within the context of the TALON's diverse use cases, which include UC1 (Automatic UATVs Coordination), UC2 (I5.0 Automation and Planning), and UC3 (AR/VR for Training and Maintenance), the following paragraphs introduce the identified stakeholders for the TALON ecosystem. From the table below, we observe that the TALON ecosystem involves four technical roles with diverse responsibilities.

Table 2 - Stakeholder Categories

Categories	Description
First Responders or Police Officers (UC1)	First responders are individuals who are trained and equipped to be the first to arrive and aid in emergency situations. In the context of TALON, Firefighters can use the outcome of UC1 to monitor and promptly respond to fires, hazardous materials incidents, and other emergencies. Police Officers may respond to emergency calls, accidents, and incidents reported by the public. This can involve providing first aid or responding to emergency situations.
Manufacturing Employees (UC2; Scenario 1 and UC4)	Manufacturing employees are individuals who work in the manufacturing industry, which involves the production of goods on a large scale. They play a critical role in the creation, assembly, and packaging of various products across different sectors. Their key activities in TALON include quality control, production and assembly, safety, and compliance.
Optical Network Engineers (UC2; Scenario 2)	Optical Network Engineers are professionals who design, implement, and maintain optical computer networks for

	organisations. They are responsible for ensuring the smooth and efficient operation of network infrastructure, connectivity, and services.
In-Training Machine Operating Personnel (UC3; Scenario 1)	In-Training Machine Operating Personnel refers to individuals who are undergoing training to become proficient in operating specific machines or equipment within a manufacturing or industrial setting. These individuals are typically learning the skills and knowledge required to safely and effectively operate machinery used in production processes.
Remotely Connected Maintenance Expert or Personnel without maintenance expertise (UC3; Scenario 2)	A Remotely Connected Maintenance Expert, also known as a remote maintenance technician or remote support specialist, is an individual who provides technical assistance, troubleshooting, and maintenance services to clients or organisations from a remote location. Personnel without maintenance expertise will get guidance by the remote maintenance technician to resolve on-site issues.

The functionalities of the TALON System and the interactions of its core services with the TALON actors (i.e., user roles and stakeholders) are presented using UML Use Case Diagrams. The Use Case Diagram below depicts who are the targeted users of the system and how these users interact with the system. During requirements gathering, we identified different user roles, including:

- End user of a service / application
- Service developers / providers
- E2C infrastructure providers

and different stakeholders, including:

- First Responders (UC1)
- Police Officers (UC1)
- Manufacturing Employees (UC2; Scenario 1 and UC4)
- Optical Network Engineers (UC2; Scenario 2)
- In-Training Machine Operating Personnel (UC3; Scenario 1)
- Remotely Connected Maintenance Expert or Personnel without maintenance expertise (UC3; Scenario 2)

We adopt the swim lanes concretisation to indicate processes which generate data to the TALON System, processes which ensure the secure interaction of APIs, services and pods, as well as processes which train AI Models to both support the end user applications and the intelligent mechanisms of the E2C AI-fuelled Orchestration. The TALON end users have different views and thus interactions with the TALON System and therefore they provide or access different functionalities at the edge, intermediate (i.e., secure Gateway API) or cloud lanes.

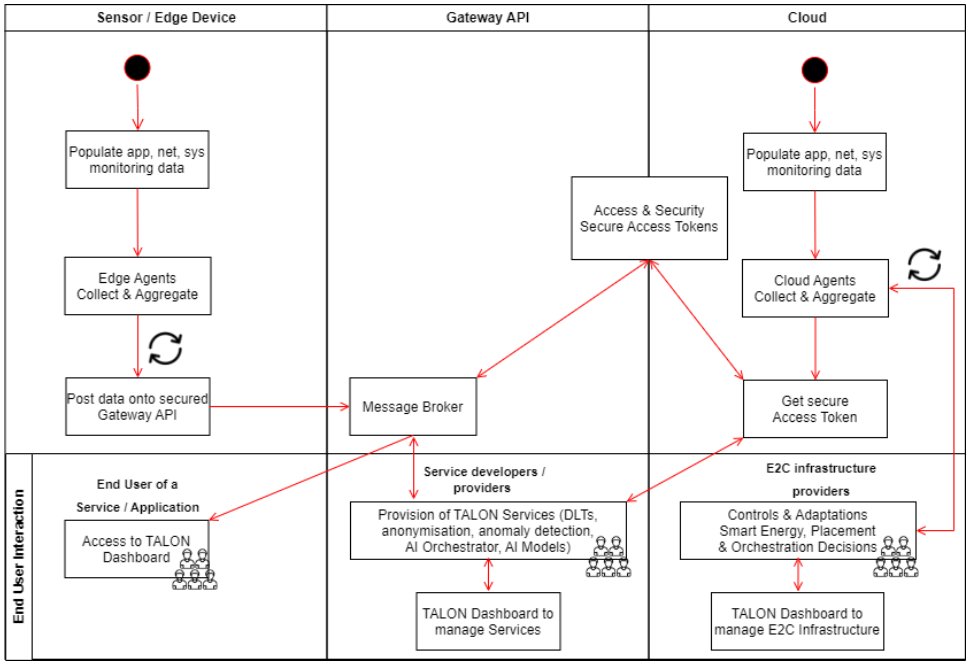

Figure 2 - TALON's UML Diagram, User Roles and Interaction with Processes

As it is depicted through the UML diagram, the TALON System can be used by the user roles to serve the entire E2C lifecycle, while at the same it can be used by the stakeholders in an agnostic manner to get access and be facilitated by the different TALON's state-of-the-art technology offerings.

3 Access and Security

The Access and Security Layer facilitates a collection of security mechanisms that are designed to protect the confidentiality, integrity, and availability of data and systems. These mechanisms in TALON will include:

- Authentication: This service will verify the identity of a user or device before granting access to a system or resource.
- Authorization: This service will determine what resources a user or device is allowed to access and what actions they are allowed to perform.
- Access control: This service will enforce the rules that have been defined for authentication and authorization.
- Identity and Access Management: This service will manage user identities and access rights to the different E2C services of the TALON System.
- Data Anonymisation: This set of services will remove Personally Identifiable Information (PII) from data by anonymising names (i.e., strings), phone numbers (i.e., numbers), and faces (i.e., images) from data sets collected and used in the TALON System.
- Anomaly Detection: This set of services will monitor network traffic for indicators of malicious activity.
- Distributed Ledgers: This set of services will support security, traceability and transparency of the federated models' weights.

3.1 Authentication and Authorisation

3.1.1 Business Logic

The role of the Authentication and Authorisation component is horizontal to the TALON System. It is responsible to decentralise the administration of access among users, devices (i.e., edge, and cloud based) and applications / data. Therefore, it is responsible to support services for secure and distributed collaboration within the TALON E2C System, credentials federation, policies specification and enforcement. It will provide a cross- layer and services infrastructure which will manage enhanced and distributed access control over the heterogeneous cloud, and edge nodes taking into consideration specificities regarding subject attributes (i.e., who accesses and how), environmental attributes (i.e., where is the origin of the request, edge, cloud and how e.g., browser, pod/agent, producer/consumer process, etc.), resource and action attributes (who is the owner, what time created, etc.). The Authentication and Authorisation component will define a set of different policies to realise all the different access requirements, time-sensitive permissions and the geographically distributed nature of components in the TALON E2C System. It will be developed abstractly to enforce the respective policy / rule. Also, it will define multiple attributes, and give the same role different permissions depending on their current security context. For example, an end user working on their secure cloud infrastructure could have one level of permissions, while the same end user connecting from her edge device will have different permissions. The Authentication and Authorisation component will allow to combine all the different attributes and criteria to create complex policies and rules. An authorisation engine will be responsible to evaluate and produce the respective policies which allow / deny under specific rights the operation requested by the role and the attributes of the end user.

At the same time, all components, systems and services or even outside (third parties) will be seamlessly certified through a gateway API to harmonize their authentication and authorization to the TALON System. The Authentication and Authorisation component will configure and provide:

- A Certificate Authority (CA) server.
- Security Configurations (SCs).
- A Gateway API to manage and forward clients traffic to backend services.
- A User Authorization and Authentication (UAA) server.

The CA is a Public Key Infrastructure (PKI) certificate authority software package that is managed and sponsored by Sectigo Limited[6]. The UAA is also an open-source Identity and Access Management solution provided by Keycloak[7] aimed at modern applications and services. It facilitates the security and administration management of access tokens among applications and services. UAA component will be deployed as docker container in UBITECH's infrastructure (which is certified with ISO 27001) and exposed to the internet via the KarkenD[8] Gateway API.

3.1.2 Functional Description

Table 3 - Authentication and Authorisation functional description

Component ID	TALON_AS_AA
Related Functional Requirement	FR01
Component Name	Authentication and Authorisation
Description	The authentication and authorisation component decentralises the administration of access among users, devices and applications / data. It develops a horizontal infrastructure to the TALON System which manages enhanced and distributed access control over the heterogeneous edge-to-cloud computing nodes taking into consideration specificities regarding subject, environmental, resource and action attributes.
Inputs & Pre-conditions	<username, password>
Outputs & Post-conditions	<attributed access token>
API	https://www.keycloak.org/docs-api/22.0.0/rest-api/
Implementation Technology	Keycloak provides an open-source software solution designed to provide single sign-on access to applications and services. It allows users to authenticate once and access multiple applications without needing to re-enter their credentials.

[6] https://www.sectigo.com/
[7] https://www.keycloak.org/
[8] https://www.krakend.io/

	KrakenD offers a stateless, distributed, high-performance API Gateway which acts as a proxy that forwards clients to backend services.

3.1.3 Information Flow and Internal Architecture

The technologies which will be used are as follows:

- User Authentication and Authorization – Keycloak framework.
- Firewall – PfSense.
- Gateway API – KrakenD framework.

The high-level architecture related with the Authentication and Authorisation component is depicted in Figure 3.

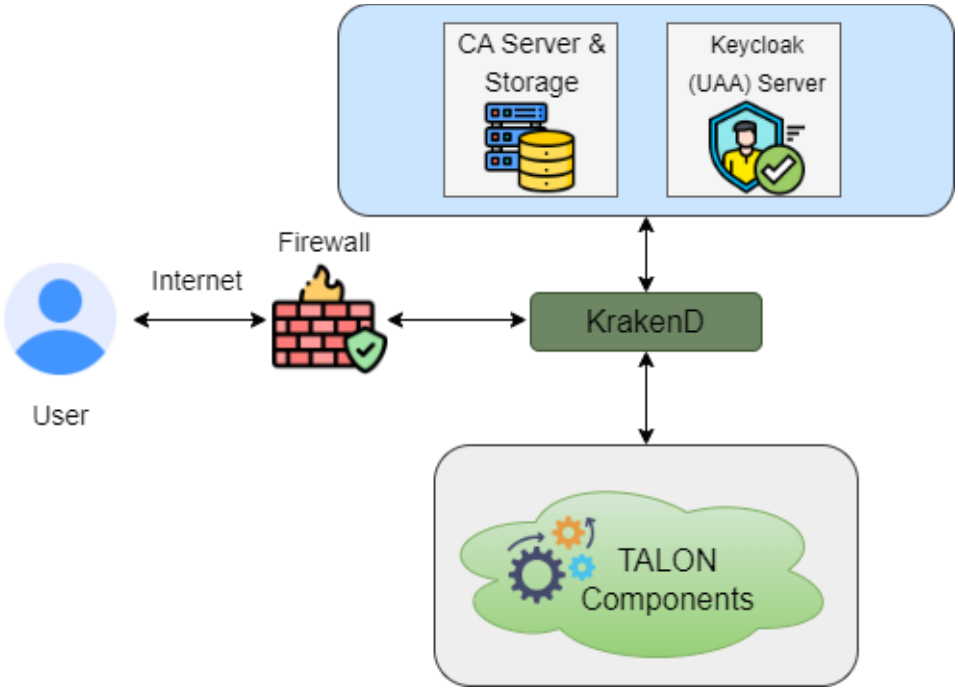

Figure 3 - Authentication and Authorisation component architecture

3.1.4 Interaction with Other Components

The Authentication and Authorisation component interacts with the following components.

Table 4 - Authentication and Authorisation interactions

Interacting Component	Description
TALON Dashboard	The Authentication and Authorisation component supports the mechanisms for secure interaction with the front-end web services comprising the Dashboard.
ALL TALON components and third-party services	The Authentication and Authorisation component supports the single sign-on (SSO) mechanisms to exchange information and certification tokens for secure

	interaction among all the TALON components and third-party services.

3.2 Data Anonymisation

3.2.1 Business Logic

The TALON project recognizes the pressing need for privacy protection in the face of the global proliferation of the Internet of Things (IoT) and the promising applications of Industry 4.0/5.0. With the importance of upholding privacy regulations, such as GDPR, in the gathering and processing of private information, the decision to deploy a data anonymisation system within TALON becomes imperative. This data anonymisation component aims to ensure secure information sharing, compliance with privacy regulations, and effective countermeasures against cyber threats.

The primary objective of this data anonymization technology is to effectively remove Personally Identifying Information (PII) from both structured and unstructured data formats. Utilising textual and numerical techniques, such as machine learning and Natural Language Processing (NLP) that can identify and anonymize common PII elements like names, emails, and phone numbers, personal data are being protected. The focus lies in anonymizing free-form text documents, which often contain unstructured data. The system employs sophisticated techniques to locate and anonymize text structures representing entities while ensuring the data's analytical value is preserved. It adheres to well-defined methods to maintain consistency and compliance with privacy laws. In the case of images, these are analysed and, if necessary, anonymized removing the related metadata.

The data anonymization component is also equipped with a module for face anonymization. Face anonymization is crucial for protecting individuals' privacy in images. By obscuring or altering facial features, it ensures compliance with privacy regulations, fosters trust, and enables the ethical use of images in the TALON contexts. It's an essential tool in upholding ethical standards and legal compliance in today's data-driven world.

The integration of these data anonymisation components, leveraging AI technologies mentioned above, yields an array of significant benefits for the TALON project. It enhances privacy protection, fostering trust and confidence among partners who share their data within the project ecosystem. The anonymisation process enables secure cross-organisational collaborations, encouraging data sharing for collective insights without compromising individual privacy. By obscuring individual identities, data anonymisation also mitigates the risk of insider threats, making internal data breaches less attractive. The system achieves a delicate balance between privacy and data utility, retaining the analytical value of data while ensuring individual identities remain protected. As a result, TALON exemplifies its commitment to ethical data handling and responsible AI practices.

The implementation approach follows a structured methodology, encompassing data pre-processing, PII recognition, anonymisation techniques, entity identification, standardised procedures, and scalability considerations to efficiently process large datasets.

3.2.2 Functional Description

Table 5 - Data Anonymisation functional description

Component ID	TALON_AS_DA
Related Functional Requirement	FR02
Component Name	Data Anonymisation

Description	The Data Anonymisation component employs advanced techniques to safeguard privacy and utility, ensuring compliance with regulations and fostering trust in secure data sharing.
Inputs & Pre-conditions	**Numerical & Text Anonymisation** 1. Data 2. AI model selection **Face Anonymisation** 1. Data containing Faces. 2. Configuration set (to define a Retention Period in case of pseudonymization, to define the wanted anonymization technique (Black Masking / Blurring / Pixelation), to choose the desired face detection technique (Haar cascades / Deep Natural Network for face detection (DNN) /...) and so on. Note: a pre-tuning of the face anonymization module is necessary to permit a coherent configuration of the anonymization module
Outputs & Post-conditions	**Numerical & Text Anonymisation** 1. Anonymised Data 2. Recognised entities **Face Anonymization** 1. Anonymised Data **2.** Recognised entities
API	**Numerical & Text Anonymisation** <under development>/ <under analysis> **Face Anonymization:** - REST API: HTTP POST/GET/PUT/DELETE
Implementation Technology	**Numerical & Text Anonymisation** 1. Python Programming Language 2. TF/Keras 3. PyTorch **Face Anonymization** 1. Docker 2. Python Programming Language 1) Pillow (ExifTags) 2) OpenCV 3) Flask

3.2.3 Information Flow and Internal Architecture

The information flow of the component is depicted in the following figures.

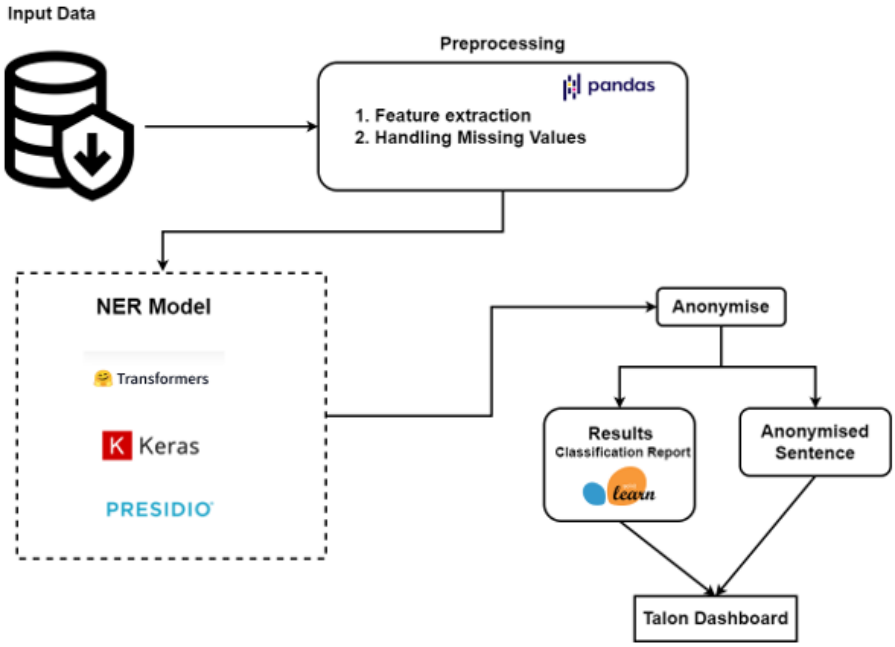

Figure 4 - Numerical and Text Anonymisation architecture

As shown in Figure 4**, Numerical and Text Anonymization component,** the anonymisation workflow begins by taking the component input, which comprises the data to be anonymised. The first step involves preprocessing the data to bring it to the required format, Once the preprocessing is finalised, advanced AI models are applied to identify and recognise entities within the dataset and to effectively anonymise the data and protect individuals' privacy. Entities refer to specific individuals, organisations, or other elements such as phone numbers, email and etc. Finally, the output of the workflow is the anonymised dataset, devoid of personally identifiable information, while maintaining its utility for various applications.

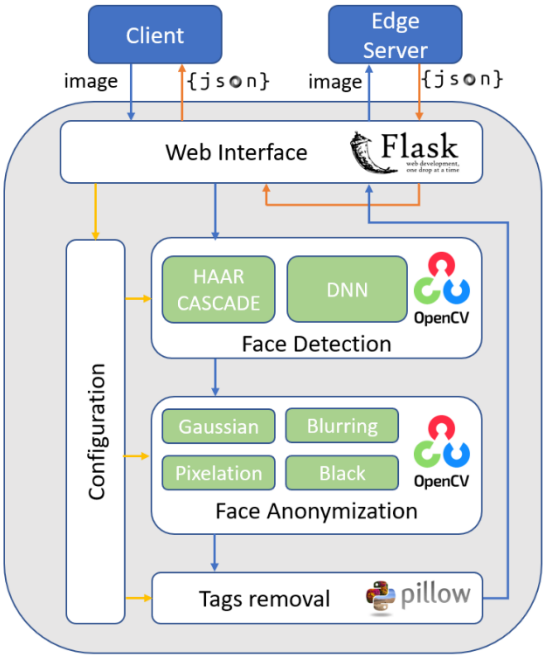

Figure 5 - Face Anonymisation architecture

In Figure 5**, Face Anonymization architecture**, is shown the workflow for face anonymization. The data received is first analysed to check if faces are present in the processed image/video, after that they are extracted from file and processed to obfuscate them. At this point the blurred faces are copied into original file. The data is considered anonymized when also the metadata associated to the file containing the faces obfuscated is removed, as it could contain personal information.

3.2.4 Interaction with Other Components

The Data Anonymisation component interacts with the following components.

Table 6 - Data Anonymisation interactions

Interacting Component	Description
TALON Dashboard	The Data Anonymisation component will communicate with TALON Dashboard to display the anonymised data and the recognised entities.
Polyglot Data Management	The Data Anonymisation component will communicate with the Polyglot Data Management to retrieve and store data for anonymisation.

3.3 DLTs for Securing AI/ML models weights

3.3.1 Business Logic

The Distributed Ledger Technology (DLT) component plays a crucial role in the security and privacy of the TALON framework. The DLT component ensures that sensitive data, such as the weights of AI models, and various evaluation metrics, are securely stored in the blockchain, a technology known for its immutability and distributed nature.

The blockchain technology provides an immutable distributed ledger enabling the integrity of the AI models weights and evaluation metrics. In the blockchain, each block is cryptographically chained with the previous one, which enables both the immutability and traceability features. These features allow authorised users to trace back and verify any changes made to the weights and the accuracy of TALON's AI models, promoting transparency within the TALON framework.

Moreover, the DLT contributes significantly to the privacy of the TALON framework by utilizing a permissioned-based blockchain technology. Such frameworks are used to verify that only authorized users with the appropriate credentials can access the data stored in the distributed ledger. This access control mechanism adds an extra layer of protection to the TALON framework, ensuring that sensitive information remain confidential and only accessible to those who have the necessary authorization.

Additionally, a lightweight version of the permissioned-based blockchain technology is used to optimize the efficiency of the DLT component, ensuring that the operational costs of the blockchain technology are minimized, and the system is scalable.

In blockchain technology trust is established through several means including the decentralized nature, smart contracts, cryptographic security, and consensus mechanisms. At first, the decentralized nature of the blockchain enhances trust as the blockchain is not relied on a central authority. Second, smart contracts play a crucial role in automating agreements and eliminating the need for intermediaries. Third, cryptographic security is heavily adopted in the blockchain as encapsulates robust encryption, digital signatures, and hash functions to provide data integrity. Last

but not least, consensus mechanisms are utilized to validate the transactions by the members of the blockchain.

3.3.2 Functional Description

Table 7 - DLT functional description

Component ID	TALON_AS_DLTS
Related Functional Requirement	FR03
Component Name	Distributed Ledger Technology
Description	Distributed Ledger Technology component is designed to provide the necessary framework for robustness and integrity to the platform, protection and trustworthiness to the E2C network, as well as a tracking of operational data. In particular blockchain is developed to offer protection, improved performance, support and secure scalability to the AI Capabilities and Transfer Learning component.
Inputs & Pre-conditions	1. Raw data 2. AI model weights 3. AI model general metrics 4. AI model evaluation metrics
Outputs & Post-conditions	1. Data monitoring 2. Secure Client Scalability
API	Under Development
Implementation Technology	1. Hyperledger Fabric 2. GO

3.3.3 Information Flow and Internal Architecture

The information flow of the Distributed Ledger Technology component is depicted in Figure 6.

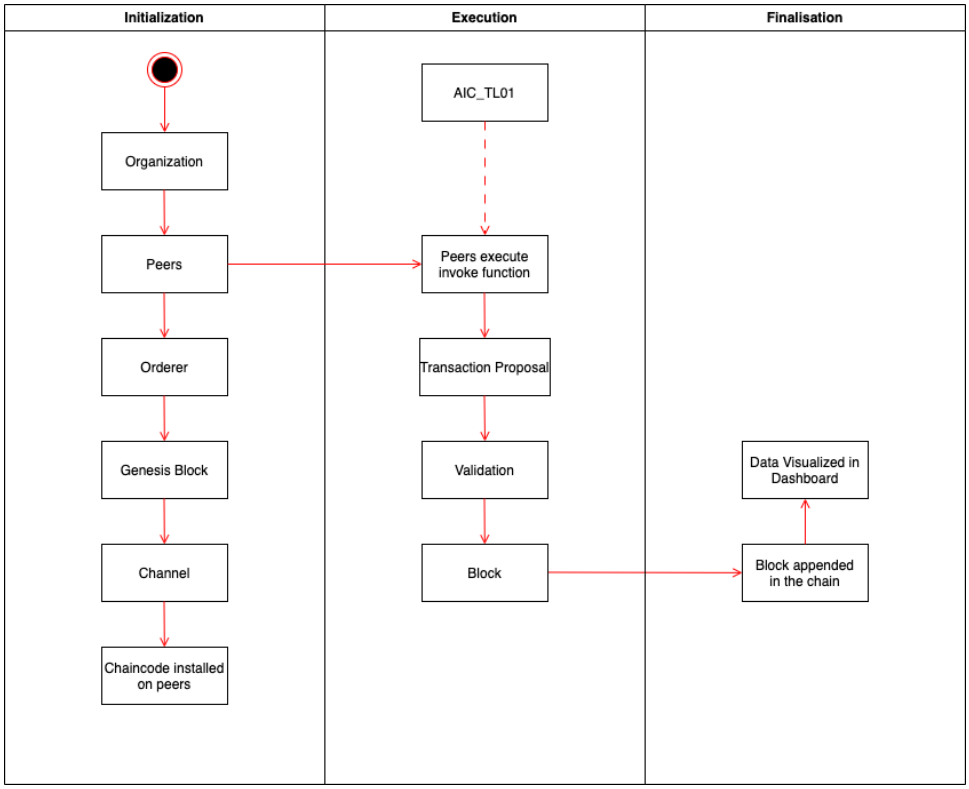

Figure 6 - Distributed Ledger Technology component architecture

TALON's lightweight blockchain component is connected but not entangled with the AI Capabilities and Transfer Learning component, as displayed at Figure 6. Each client is member both in the federated learning process and in the blockchain network, requests a transaction proposal in the blockchain.

TALON's blockchain component initialises the required entities such as organisations, peers, orders, the genesis block, channel and installs the chain code on selected peers. The blockchain component receives as input information from the AI Capabilities and Transfer Learning component in JSON format. AIC_TL01 provides several important metrics of the AI operations such as the evaluation metrics of the model and robustness metrics of TALON System. Moreover, the ML model weights and the timestamp of the training and/or the evaluation of the algorithms can be stored in the ledger.

Once a training round in the Federated Learning (FL) process finish, the smart contract is triggered, and new transaction proposals are made to the blockchain and the verification process in the blockchain begins. The transactions proposals which include the information from the FL process, undergo verification before included in a block. Finally, the block is chained with the previous block.

3.3.4 Interaction with Other Components

The DLTs for Security and Privacy component interacts with the following components.

Table 8 - DLT interactions

Interacting Component	Description
AI Capabilities and Transfer Learning Component	The Distributed Ledger Technology component will communicate with the AI Capabilities and Transfer Learning component to store critical information of the FL

	process such as AI model weights, and evaluation metrics.

3.4 Anomaly Detection

3.4.1 Business Logic

Networking and distributed computing have grown to be fundamental components of contemporary automation systems since the advent of Industry 4.0 (I4.0). The adoption of greener, lighter and more efficient hardware and software is growing in manufacturing environments, since they improve the efficiency of tasks overall. As manufacturing systems' complexity keeps improving and their potential for attack expands, it is harder to provide security as well as effective performance in the industrial systems [7]. TALON's security monitoring and analysis component aims at providing the necessary landscape for efficient protection against malicious networking activities without sacrificing the platform's performance. Utilising efficient feature engineering and lightweight yet very performing algorithms, TALON's anomaly detection component has the capacity to analyse communication protocols and providing the corresponding security events.

On top of the automated AI analytical operations, potential adversarial activities can also be identified and mitigated through visualisation mechanisms [8]. Utilising visualisation techniques along with the AI has significant added value, as experts can recognise and confirm insights without the need of a ground truth. Employing the human-in-the-loop factor in the decision-making process, renders TALON system to be more trustworthy and efficient through human-machine collaboration activities. Utilising this hybrid approach for anomaly detection, end users and industrial stakeholders can sustain high predictability and detection for protocol and datasets with high complexity.

By implementing ML algorithms along with visualisation techniques, TALON's Anomaly Detection module offers administrators a holistic anomaly detection solution. The AI section of the module comprises of the feature engineering, the training and the detection parts offering an "adaptive" solution tailored to the specific protocol or dataset. AI enabled anomaly detection solutions are mandatory, as diverse pilot networks have their own distinctive peculiarities. The different characteristics of each pilot network can be better captured utilising AI enabled anomaly detection, instead of static rule-based monitoring.

Among other things, a pilot's network may comprise of VMs, PCs, smart devices, and workstations. These devices communicate with each other forming a network, upon which the Anomaly Detection module will be implemented. The pilot network may have various standardised communication protocols depending on the requirements and the scope of the task. Such protocols are HTTP, DNP3, TCP, UDP, IEC and many more. Due to the variety of the protocols and their features, TALON's Anomaly Detection Module aims to provide useful information to the end user, reducing the number of features to the most important ones, thus providing less complexity and more efficiency to the visual representations of the platform.

3.4.2 Functional Description

Table 9 - Anomaly Detection functional description

Component ID	TALON_AS_AD
Related Functional Requirement	FR04

Component Name	Anomaly Detection
Description	TALON's AI visual-based anomaly detection component will have the capacity to watch and identify severe cyber security attack types. Moreover, this application can create reports and easily understandable graphs that provide an accurate overview of the network status.
Inputs & Pre-conditions	Network traffic
Outputs & Post-conditions	1. Security Event logs 2. Intuitive Graph
API	Under Development
Implementation Technology	Python

3.4.3 Information Flow and Internal Architecture

The information flow of the AD component is depicted in Figure 7.

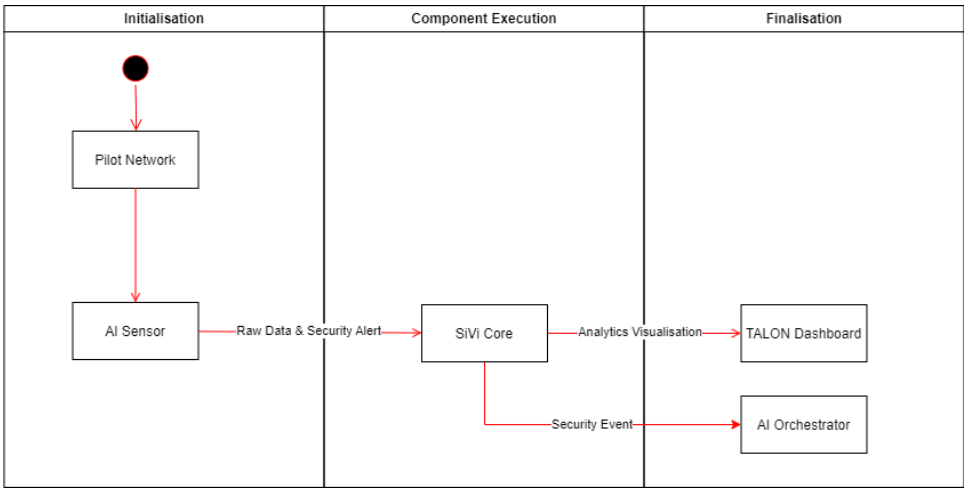

Figure 7 - Anomaly Detection component architecture

TALON's Anomaly Detection Module will have a series of capabilities rendering it a robust threat detection engine. The first step is to provide the necessary network packets to the system. A sensor will be able to detect the network traffic from the pilot network and generate the corresponding response depending on whether the traffic is malicious or benign. Then, the sensor communicates with the SiVi Core sub-component and sends the corresponding raw format of the detected malicious activity along with the AI results. SiVi core utilises some advanced technologies to unify and prepare the gathered information so the results can be communicated and illustrated near-real time and historically. In first scenario, the user can check the data flows as they are monitored, described and predicted, while in the second scenario the user can select a time window to undertake for analysis.

3.4.4 Interaction with Other Components

The Anomaly Detection component interacts with the following components.

Table 10 - Anomaly Detection interactions

Interacting Component	Description
TALON Dashboard	The Anomaly Detection component will interact with the TALON Dashboard to report back the security event logs of the detected malicious network flows (if any).

4 AI-fuelled Orchestration

This section investigates the process of coordinating dynamic interactions within the TALON framework. It examines the design and operation of several fundamental components that manage orchestration aspects of the system. These components cover a broad spectrum of essential functions, including service modelling and enactment, next-generation Software-Defined Networking, distributed intelligence, and AI swarm orchestration.

4.1 Application Lifecycle Manager

4.1.1 Business Logic

Orchestration is one of the most difficult goals of the TALON vision and the main reasons for that are boiled down to the following facts:

- TALON embraces the “federated scheduling” concept where edge resources and cloud resources will be treated as a uniform pool of resources that are able to host cloud-native service graphs.
- This pool of resources can be shared by more than one service graphs simultaneously (i.e., multi-app deployment), with completely different activated policies; hence there is a problem of efficient resource-sharing.
- Two service-graphs that employ both edge and cloud resources simultaneously may require contradictive management actions to achieve their SLOs.

All these goals will be materialized by the Application Lifecycle Manager. More specifically, the component has the following main responsibilities:

1. Monitor, take decisions and coordinate the deployment procedure of the service graph.
2. Check if the applied SLOs are fulfilled and execute corrective elasticity actions if they are not.
3. Provide an abstraction model where corrective actions can be registered and applied as runtime adaptation (i.e., these may be elasticity actions, security actions, etc).
4. Try to maintain a consistent view of the “advertised” physical resources and their energy and reservation status.
5. Predict and resolve conflict resolutions raised by different policy-actions.

From a theoretical point of view, the Application Lifecycle Manager falls under the category of an Automatic Control System (a.k.a. ACS). As such its behaviour is organized in the form of a closed control loop that evaluates continuously the existing resources, the deployment requests, and the reconfiguration requests of the already deployed applications. The main difficulty relates to the fact that the entire control loop should remain always in equilibrium taking under consideration that each distinct application is practically an “autonomous” entity.

Taking under consideration all the above, it is self-evident that the orchestration business logic can be seen as a centralized logical entity; yet it is a physically distributed entity since it affects “scheduling” of applications-graphs and applications-individually at various levels i.e., at the cloud-level and at the edge-level.

4.1.2 Functional Description

Table 11 - Application Lifecycle Manager functional description

Component ID	TALON_AIO_ALM
Related Functional Requirement	FR05, FR11
Component Name	Application Lifecycle Manager
Description	The application lifecycle manager component utilizes AI-fuelled and adaptive scheduling algorithm that dynamically allocates resources on both Cloud side and Edge side. It supports the deployment of all the modern big data and ML frameworks. Its main goal is to fulfil applications needs and user-defined SLOs. This intelligent scheduling approach ensures that applications receive the necessary resources to operate optimally while maintaining fairness among competing workloads.
Inputs & Pre-conditions	Application Graph; Service Level Objectives
Outputs & Post-conditions	Application Status
API	http://talon-backend.euprojects.net/applicationmanager/swagger-ui/
Implementation Technology	Programming Language: Java, Frameworks: Spring boot, Frabric8 Custom Spring boot application translates application graph to YAML structure. Talon Scheduler is built on top of Fabric8 library which ensures the end-to-end integration (i.e., deployment and management) between cloud-native applications and Cloud/Edge infrastructure

4.1.3 Information Flow and Internal Architecture

The high-level architecture related with the Application Lifecycle Manager Component is depicted in Figure 8.

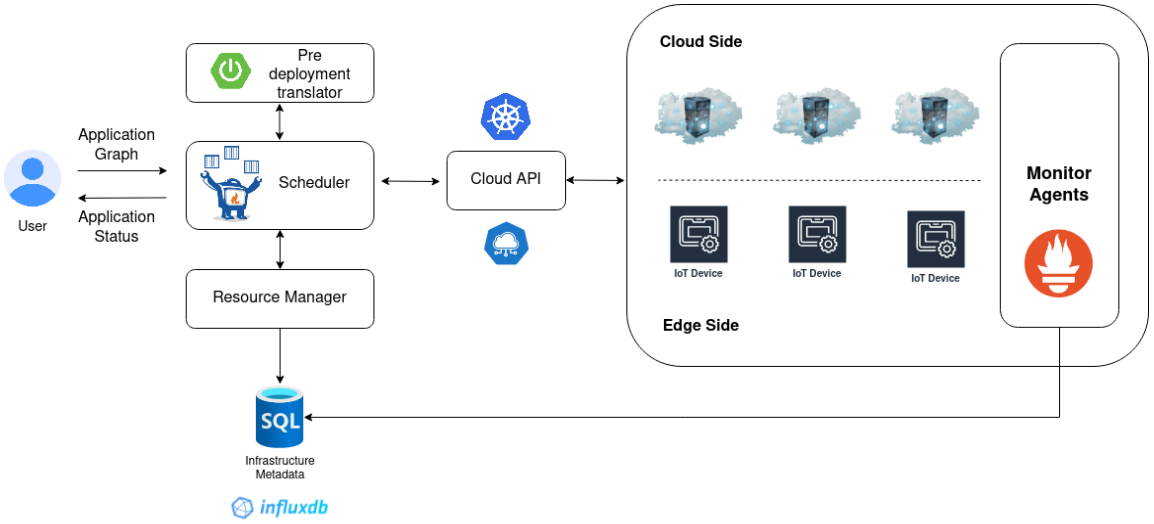

Figure 8 - Application Lifecycle Manager Component architecture

4.1.4 Interaction with Other Components

The Application Lifecycle Manager component interacts with the following components.

Table 12 - Application Lifecycle Manager interactions

Interacting Component	Description
Smart Policy Manager	It receives input from the Smart Policy Manager that offers incentives for offloading computations in order to modify and optimize the YAML file in the meaning of performance and cost.
Resource Allocation and Deployment	It receives resources availability and constraints in order to modify and optimize the placement of the microservices in the application graph
Data Monitoring, Collection and Aggregation	It pushes aggregate application metrics to the logically centralized measurement collection component.
AI Model Training, Adaptations and SLOs	It receives hints for the future state of the Cloud/Edge infrastructure and modifies resource properties of YAML files to fulfil SLOs
TALON Dashboard	The Application Lifecycle Manager component interacts with the TALON Dashboard to report back the applications' deployment status.

4.2 NG-SDN and Distributed Intelligence

4.2.1 Business Logic

The role of the NG-SDN and Distributed Intelligence is to secure onboarding, and operation of a consistent network overlay among the edge and cloud nodes and the selection of a cluster-head which will represent an entire physical deployment. The purpose of the network overlay is to:

- Populate network metrics to feed the AI Model Training for SLOs adaptations through the Smart Policy Manager.
- Manage virtual ports, IP addresses and pods communication.
- Manage virtual overlay networks.
- Take part in the selection process of a cluster-representative (cluster-head) which will be used to offload several computational and computational tasks.

Semi-centralised techniques will be used for cluster-node selection. Each node is not statically configured with the address of the super-node. Instead prior to the joining protocol, the node is executing a cluster-head election protocol. The goal of such protocols is to logically split the network topology to several 'islands'; each of which is represented by a cluster-head. This is performed using a combination of controlled flooding schemes initiated randomly by volunteering-nodes based on their capabilities (battery, memory etc). Nodes within a specific proximity that receive these broadcasts and have no super-node accept the cluster-head as their super-node. Hence, during joining and leaving they require their 'topological position' from the elected node.

4.2.2 Functional Description

Table 13 - NG-SDN and Distributed Intelligence functional description

Component ID	TALON_AS_SDNDI
Related Functional Requirement	FR06
Component Name	NG-SDN and Distributed Intelligence
Description	The core functionality of the component is to manage virtual overlay networks between edge and cloud. The input of the component is Network Rules that update routing or replacing a pod in the cluster on runtime. The component feeds itself with new metrics to optimize the deployments based on the network perspective.
Inputs & Pre-conditions	Network Rules
Outputs & Post-conditions	Network Metrics
API	http://talon-backend.euprojects.net/sdn/swagger-ui/
Implementation Technology	Programming Language: Python Frameworks/Libraries: Kubeovn, Flask

4.2.3 Information Flow and Internal Architecture

The high-level architecture related with the NG-SDN and Distributed Intelligence component is depicted in Figure 9.

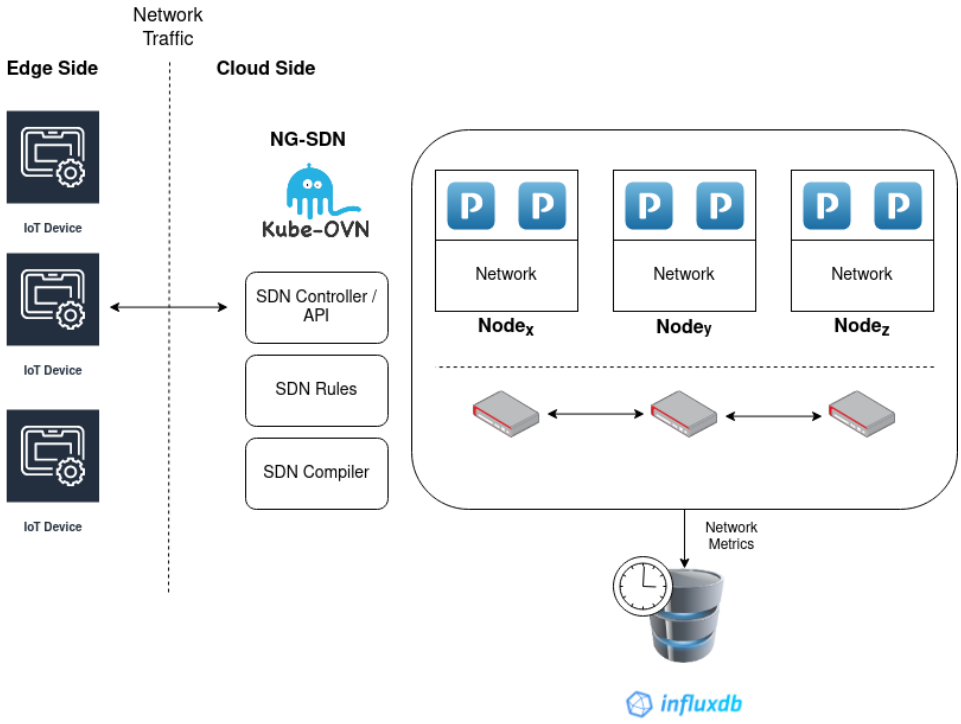

Figure 9 - NG-SDN and Distributed Intelligence Component architecture

4.2.4 Interaction with Other Components

The NG-SDN and Distributed Intelligence component interacts with the following components.

Table 14 - NG-SDN and Distributed Intelligence interactions

Interacting Component	Description
Application Lifecycle Manager	It can propagate specific events that cannot be handled within the vicinity of the agent.
Data Monitoring, Collection and Aggregation	It pushes aggregate network metrics to the logically centralized measurement collection component.
TALON Dashboard	The NG-SDN and Distributed Intelligence component interacts with the TALON Dashboard to report back the network status of the E2C infrastructure.

4.3 AI Swarm Orchestration

4.3.1 Business Logic

The role of the AI Swarm Orchestrator is to compute the vectors, manage and organize the trajectories and positions in 3D space for all involved drones in-real-time. The idea is how to utilize a look-ahead mechanism that enables all drones in a swarm to project their next moves, like a map, so that they lower energy consumption and have no need to request for re-evaluation at every step of the process.

Key features:

- Environment aware system
- Integrates with various sensors in case of extension
- Adaptable to changes (location and space-wise)

4.3.2 Functional Description

Table 15 - Swarm Orchestration functional description

Component ID	TALON_AIO_AISO
Related Functional Requirement	FR07
Component Name	DRAGONFLY DF-X
Description	DF-X is a specialized AI software and subsystem of the AiRFLOW platform. It enables real-time collision detection & collision avoidance (CD/CA) for autonomous and unsupervised flights.
Inputs & Pre-conditions	Actual position and location of master drone (1 drone) and relative of the rest
Outputs & Post-conditions	Map of trajectories and positions
API	Open API available
Implementation Technology	Python, Linux

4.3.3 Information Flow and Internal Architecture

The information flow of the component is depicted in the following Figure 10.

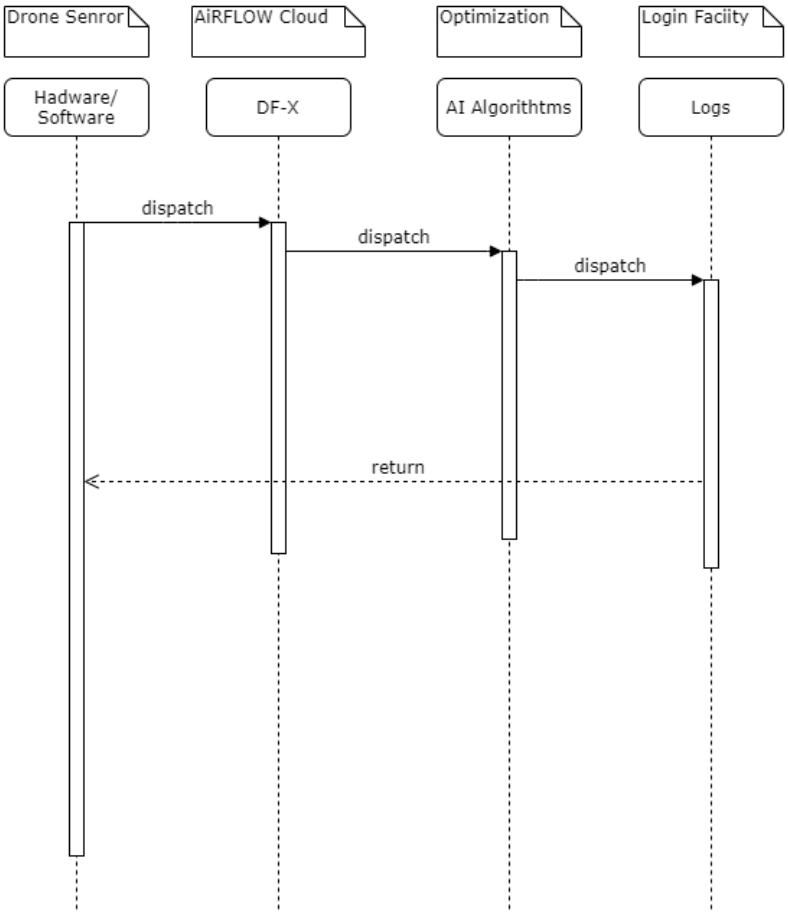

Figure 10 - Swarm Orchestration sequence diagram

The DRAGONFLY component exists inside the drone or at the edge mobile suitcase (called TALOS) that interconnects the drone with the AiRFLOW ecosystem (for more information check section 4.1 of TALON D2.1). TALOS requests inputs from the drone sensors (IMU, GPS, Accelerometer and battery information) and forwards this information to the AiRFLOW cloud where the DF-X (the orchestrator) lives. The DF-X then organizes the vectors and positions of each drone and sends back to TALOS a vector-position map. Then, TALOS propagates the information to all drones and drones set the new positions, altitudes and headings. Finally, an API that shares all information from DF-X is provided to the developers.

4.3.4 Interaction with Other Components

The AI Swarm Orchestration component interacts with the following components.

Table 16 - Swarm Orchestration interactions

Interacting Component	Description
TALON Dashboard	The Authentication and Authorisation component supports the mechanisms for secure interaction with the front-end web services comprising the Dashboard.
Note	The AI Swarm Orchestrator practically requires no other interactions at this stage of development.

4.4 Resource Allocation and Deployment

4.4.1 Business Logic

The resource manager component of TALON has as role to receive input from the E2C orchestrator regarding the available edge and cloud resources and the applications that are to be executed and decide where the tasks concerning the application should be executed and how they should be scheduled. Decisive for the decisions are SLOs concerning each of the use cases. Hence, the resource manager component will formulate appropriate task placement and scheduling optimisation models, separately for each use case, taking into account metrics such as energy consumption and latency. The formulated problems can be solved by a combination of conventional optimization methods and also of AI based approaches for the reduction of the running time.

4.4.2 Functional Description

Table 17 - Resource Manager functional description

Component ID	TALON_AIO_RAD
Related Functional Requirement	FR08
Component Name	Resource manager
Description	Based on the particular SLAs from the use cases and the availability of resources, the resource manager decides to which layer a task will be executed, i.e., mobile device, edge, or cloud, and how the tasks will be scheduled in the particular layers after the offloading decisions have been made
Inputs & Pre-conditions	i) SLAs of the use cases; ii) availability of resources
Outputs & Post-conditions	Offloading and task scheduling decisions
API	Under development. Possibility to use REST API
Implementation Technology	Mainly Python with possibility of also combining it with Matlab

4.4.3 Information Flow and Internal Architecture

The TALON resource manager block with its components is shown in Figure 11.

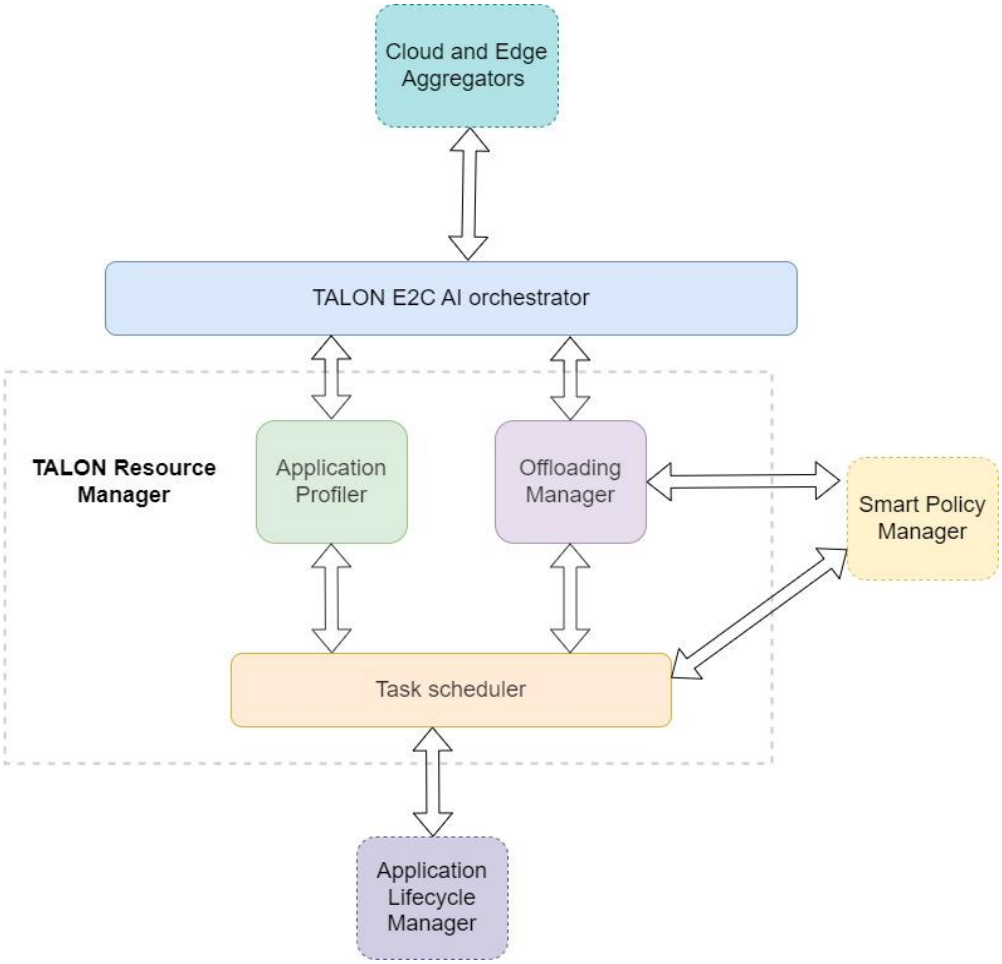

Figure 11 - Resource Manager architecture

The functionality of the components of the resource manager is as follows:

i) ***Application profiler***: It receives the applications from the E2C orchestrator and partitions them into modules (subtasks) which can be processed through parallel execution. It also profiles the resource requirements (computational, energy, and data transfer requirements) of the particular modules.

ii) ***Offloading manager***: It decides which modules will be offloaded and where the offloading will take place. The formulation of the offloading problem is normally of a mixed-integer linear programming (MILP) nature, which are NP-hard problem. Hence, based on their complexity, their solution will involve either heuristics or ML-based approaches.

iii) ***Task scheduler***: Once the offloading has taken place, the task scheduler decides how the tasks will be scheduled for execution in the corresponding layer. As in the case of the offloading problem, the task scheduling problem is normally a MILP problem. Hence, heuristics together with ML methods will be used for its solution.

4.4.4 Interaction with Other Components

The Resource Allocation and Deployment component interacts with the following components.

Table 18 - Resource Manager interactions

Interacting Component	Description
Smart Policy Manager	The **resource manager module** of TALON will receive input from the **smart policy manager** regarding pricing schemes that will be leveraged by the offloading manager component for deciding to where modules will be offloaded. In turn, it will also provide input to this component regarding the nodes that successfully executed the tasks that can be leveraged for updating the pricing models.
Edge and Cloud Aggregators	The resource manager component of TALON will receive input, through the E2C orchestrator, from the edge and cloud about the availability of the cloud and edge resources.
Application Lifecycle Manager	The outputs of the **resource manager module (offloading and task scheduling decisions)** will be fed to the **application lifecycle manager** module that will be used for the continuous evaluation of the existing resources. In addition, failure of modules in task execution will be registered and will be forwarded to this component for corrective actions to be taken (selection of alternative devices for task execution).

4.5 Smart Policy Manager

4.5.1 Business Logic

This component utilises the analytics computed from monitoring the edge and cloud services to compute smart policies. The objectives to be met have been defined as SLOs along with the respective metrics (i.e., computing, energy and cost) to realise them. The AI model training provide input to the Smart Policy Manager to enforce the runtime adaptations. Then, the Smart Policy Manager interacts with the Application Lifecycle Manager to re-configure or re-deploy such adaptations.

4.5.2 Functional Description

Table 19 - Smart Policy Manager functional description

Component ID	TALON_AIO_SPM
Related Functional Requirement	FR09
Component Name	Smart Pricing tool
Description	This component integrates AI technologies for smart policies in TALON's edge AI network, regulating resource sharing, ensuring trust, and providing computational offloading incentives. AI techniques adapt pricing, optimise resource allocation and promote an efficient E2C framework.

Inputs & Pre-conditions	A variety of inputs will be taken under consideration, but the exact ones are now being defined.
Outputs & Post-conditions	Pricing.
API	Swagger Documentation
Implementation Technology	Programming languages: Python, Javascript, SQL, mySQL Frameworks/ libraries: Tensorflow/Keras/Scikit-Learn, numpy, pandas.

4.5.3 Information Flow and Internal Architecture

The information flow of the Smart Policy Manager component is depicted in the following Figure 12.

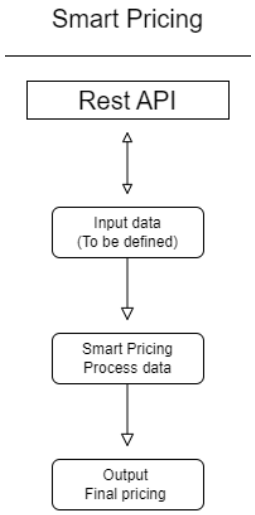

Figure 12 - Smart Pricing architecture

The internal logical architecture of this component adopts a systematic flowchart approach. It begins by collecting essential input data, such as CPU usage, battery consumption, and other relevant metrics from the connected devices within the TALON edge AI network that will be more specifically defined. This data is then subjected to sophisticated AI techniques for analysis, which may include machine learning algorithms, pattern recognition, and predictive modelling. The final output of this process is a dynamic and adaptive pricing model that caters to the specific needs and behaviours of users within the edge AI network.

4.5.4 Interaction with Other Components

The Monitoring Metrics component interacts with the following components.

Table 20 - Smart Policy Manager interactions

Interacting Component	Description
AI Model Training, Adaptations and SLOs	The Smart Policy Manager interacts with the AI Model Training to get as input the SLOs and the re-calculated metric values to translate them into YAML files for AI Orchestration re-configuration and re-deployment.
Application Lifecycle Manager	The Smart Policy Manager interacts with the Application Lifecycle Manager to enforce the calculated policies and support the runtime adaptations.

4.6 Data Monitoring, Collection and Aggregation

4.6.1 Business Logic

Large volumes of data created by multiple distributed system components are efficiently managed and analysed thanks to the Data Monitoring, Collection, and Aggregation component. This component offers a metrics pipeline that makes it possible to gather and aggregate monitoring data from many components that are stored in various places and formats.

In a distributed system like TALON, various tools generate performance metrics, energy statistics, and other essential data points. These metrics offer important information about the performance, resource use, and overall health of the system. These agents pull relevant data from the components or cluster nodes and record this information directly into a storage system exposed then to all instances of the system. This ensures a unified view of the system's performance and avoids overburdening individual components with monitoring responsibilities.

The Kubernetes environment supports monitoring solutions providing scalability, availability and all the benefit of the case. Hence, our approach for general metrics includes monitoring key parameters such as RAM, CPU usage, and GPU utilization. Additionally, the component works with a specialized solution for energy metrics, as energy consumption plays a significant role in the project. By making this monitoring data available to the application instances, they can leverage it to train their AI models effectively. Moreover, the AI models can use this information intelligently to make informed decisions about the optimal resource allocation and placement, effectively acting as a fully aware orchestrator.

A core feature of the TALON scalable and flexible monitoring stack will be its support for interoperable edge node utilization and application behaviour metric collection. Specifically, TALON monitoring stack will provide one unified and interoperable metric interface that can be fed with monitoring data by different collection mechanisms (probes). Various probes will be developed and provided by TALON, including a probe for docker container monitoring, energy monitoring, security metric extraction and network monitoring. A probe template embedding monitoring metric abstractions will be provided so that edge service users can develop their own custom probes harvesting data from their deployed services.

4.6.2 Functional Description

Table 21 - Data Monitoring, Collection and Aggregation functional description

Component ID	TALON_AIO_DMCA

Related Functional Requirement	FR10
Component Name	Data Monitoring, Collection and Aggregation
Description	Data Monitoring, Collection, and Aggregation component involves the real-time observation and tracking of various metrics, events, and activities within the E2C infrastructure. It summarizes the collected data into more manageable statistical reports in order to support the data-driven decision-making.
Inputs & Pre-conditions	Metrics Collected from E2C infrastructure
Outputs & Post-conditions	Metrics Reports
API	Swagger Documentation to be defined
Implementation Technology	Programming Language: Python Programming Framework: Flask Framework: Kepler, InfluxDB

4.6.3 Information Flow and Internal Architecture

The flow of energy metrics extracted and collected is shown in Figure 13.

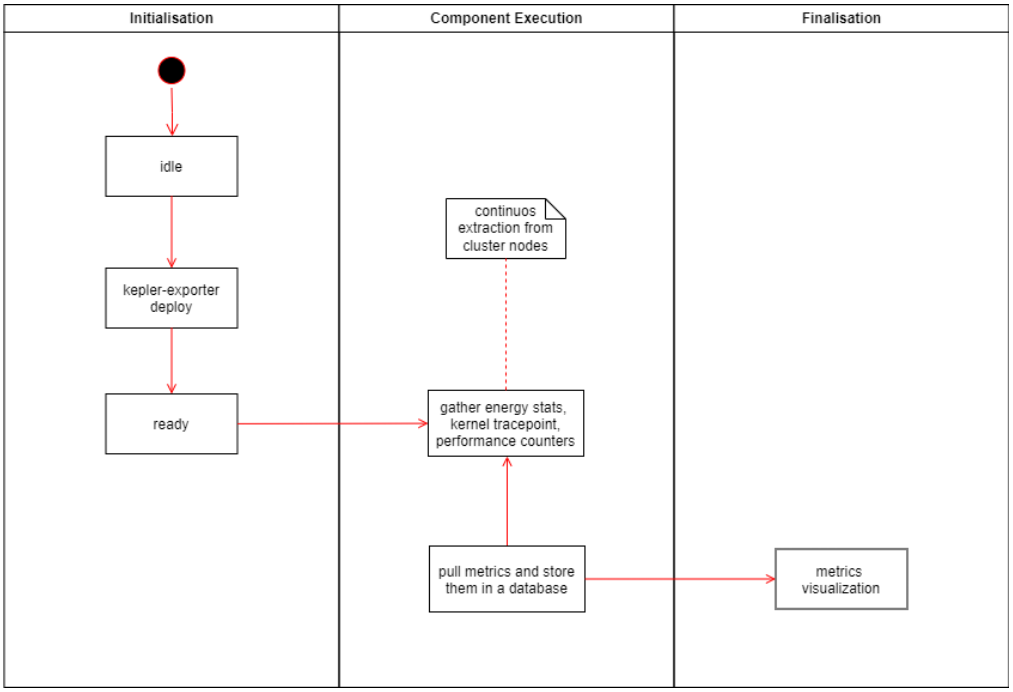

Figure 13 - Data Monitoring, Collection and Aggregation sequence diagram

The extraction of energy metrics from the various nodes in the cluster is done through the use of a framework called Kepler.

Kepler is an energy metrics exporter that uses eBPF to probe the CPU performance counters and tracepoints of the Linux kernel, eBPF is a revolutionary technology with origins in the Linux kernel that can run sandboxed programmes in a privileged context such as the operating system kernel. It is used to extend the capabilities of the kernel safely and efficiently without requiring changing kernel source code or load kernel modules.

The tool integrates with Kubernetes to collect energy information from CPU, GPU and RAM through eBPF programmes. Kepler collects kernel-level data to ensure transparency in the allocation of energy to processes, in this way exposes various metrics on the energy consumption of Kubernetes components, including pods and nodes.

This data is periodically extracted from InfluxDB (or similar databases like Prometheus) so that a history of consumption trends can be maintained. Such database is designed to collect and store time-series data, it operates as a separate component that scrapes and collects metrics from various targets such as containers, services, nodes, and other resources. It employs a pull-based model, regularly querying the endpoints exposed by these targets to gather relevant metrics. These metrics are stored facilitating querying and analysis.

Finally, it is possible to translate these metrics into a readable dashboard to help visualize energy consumption. Alternatively, these data and statistics can be used by further components of TALON from which they can build ML models predicting energy consumption by pods or components that contribute to intelligent workload scheduling decisions in the system helping to reduce energy consumption.

Figure 14 - Data Monitoring, Collection and Aggregation architecture

Figure 14 illustrates the dynamic interactions that support the extraction and presentation of energy metrics. The process starts with Kepler's integration into the Kubernetes environment. Once deployed, Kepler acts as an observer, gathering real-time metrics that provide insights into the dynamics of resource utilization.

This real-time data is stored in a dedicated time-series database, such as Prometheus. By accessing the visible metrics, Prometheus takes responsibility for data retrieval, ensuring persistent availability and reliability. This approach not only enables real-time monitoring but also establishes the basis for reviewing past data.

At the same time, a dashboard is implemented to act as a link between raw data and actionable insights. Powered by the data stored in Prometheus, the dashboard converts numerical data into visual representations, making energy consumption easy to understand.

This system combines Kepler's live monitoring, time-series data management, and the dashboard's visualisations to guide the Kubernetes environment towards optimal performance.

4.6.4 Interaction with Other Components

The Data Monitoring, Collection and Aggregation component interacts with the following components.

Table 22 - Data Monitoring, Collection and Aggregation interactions

Interacting Component	Description
AI Model Training, Adaptations and SLOs	Data Monitoring, Collection and Aggregation component gives input to the AI Model Training to enforce smart orchestration strategies and runtime adaptations.
Polyglot Data Management	Data Monitoring, Collection and Aggregation component collects data from the Polyglot Data Management component in order to produce the reports
Resource Allocation and Deploy	This component provides input to the resource allocation component, which will determine the scheduling and location of tasks.

4.7 AI Model Training, Adaptations and SLOs

4.7.1 Business Logic

Policies are expressed in the form of SLOs that have to be continuously achieved. A critical issue regarding these SLOs is that in order to be evaluated specific measurements and conditions have to be evaluated that correspond to the service components, its allocated resources and the execution context. It is not feasible for these measurements to be extracted and evaluated at the top-level. The reasons for that are multi-fold. First and foremost a single point of evaluation would assume that all measurements are exhaustively propagated to a single physical point. This would generate enormous traffic even for simple SLOs and would constitute a misuse of resources. Moreover, some measurements cannot be propagated at all since they live within the vicinity of for resources. This is the reason why we maintain collected measurements in the Cloud Aggregator and Edge Aggregator, separately. When these two entities need to be coordinated, the decision is taken by the Application Lifecycle Manager. The AI Model Training, Adaptations and SLOs component is continuously fed with streams of monitoring metrics in a structured format. These metrics are used for continuous training to compute runtime adaptations for different SLOs. These adaptations can be performed either periodically or trigger-based.

4.7.2 Functional Description

Table 23 - AI Model Training functional description

Component ID	TALON_AIO_AIMTAS
Related Functional Requirement	FR11
Component Name	AI Model Training, Adaptations and SLOs
Description	Utilizing cloud-related metrics contain temporal data in order to predict resource usage, workload demand, or performance metrics. The component explores recurrent neural networks (RNNs) and advanced forecasting algorithms to predict future values based on the historical data available in real-world applications running on the TALON platform

Inputs & Pre-conditions	Edge/Cloud Metrics
Outputs & Post-conditions	Trained models for predicting Edge/Cloud resources state
API	http://talon-backend.euprojects.net/ai/swagger/
Implementation Technology	Programming languages: Python, SQL Frameworks/ libraries: Tensorflow/Keras/Scikit-Learn, Kserve, numpy, pandas, Flask.

4.7.3 Information Flow and Internal Architecture

The high-level architecture related with the AI Model Training, Adaptations and SLOs component is depicted in Figure 15.

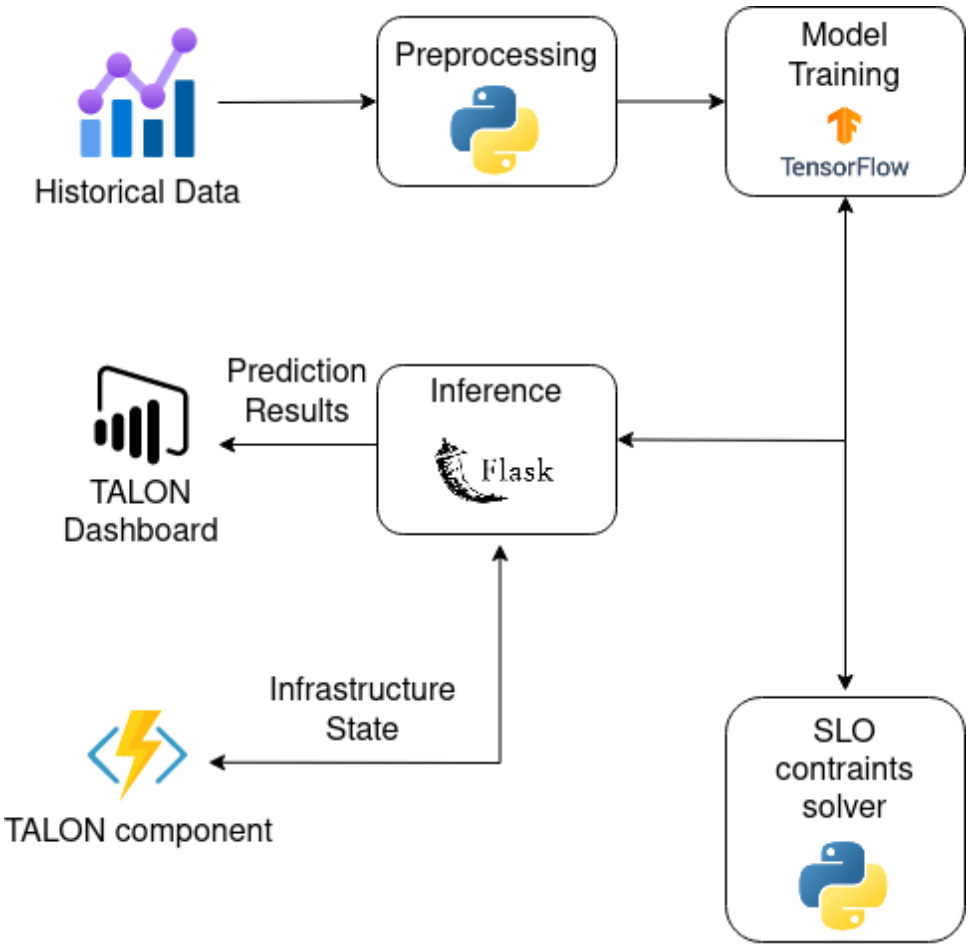

Figure 15 - AI Model Training, Adaptations and SLOs Component architecture

4.7.4 Interaction with Other Components

The AI Model Training, Adaptations and SLOs component interacts with the following components.

Table 24 - AI Model Training interactions

Interacting Component	Description
Smart Policy Manager	Interacts with Smart policy Manager in order to provide insights related to the current and future state (resources utilization) of the Edge and Cloud infrastructure so that it can optimize the computational offloading and profit model
Data Monitoring, Collection and Aggregation	Consumes historical data to train machine learning models, enabling them to learn patterns, make predictions, and gain insights that contribute to informed decision-making and improved performance.
Application Lifecycle Manager	Provides prediction related to the future state (resources utilization) of the Edge and Cloud infrastructure so that it can optimize the placement and the resources allocation

5 AI Cognition

In this section, we explore the components of the AI Cognition Layer and their key roles in TALON's intelligence, using a spectrum of generic and domain-specific analysis algorithms, digital twins, self-healing capabilities, XAI methods and data processing.

5.1 Self-healing and Self-correcting

5.1.1 Business Logic

The role of the Self-healing and Self-correcting component will be to analyse in real time the different processes on the TALON network and identify any anomalies. Any anomaly is characterised as a disruptive event, which means that any event that happens poses risk to the smooth operation of the TALON network. The second role of the self-healing and self-correcting component will be the suggestion of potential solutions that should be implemented to the network for mitigating the disruption of the corresponding event. More specifically, the self-healing and self-correcting component will adopt the thinking principles from the manufacturing domain and particularly the Zero-Defect Manufacturing (ZDM) approach, in ZDM there are four strategies that can be implemented, the detection, repair, prediction and prevention. Those strategies are implemented in pairs. There are three possible pairs, detect-repair, detect-prevent and predict–prevent. Implementing those pair strategies at the network level will allow high resiliency to the TALON network.

5.1.2 Functional Description

Table 25 - Self-healing and Self-correcting functional description

Component ID	TALON_AIC_SHSC
Related Functional Requirement	FR12
Component Name	Self-healing and self-correcting
Description	The role of this component is to analyse the TALON network, identify potential disruptive issues and suggest solutions to mitigate the issue.
Inputs & Pre-conditions	TALON network data, data from the nodes etc
Outputs & Post-conditions	Information about the status of the TALON network, issues identified and suggested solutions
API	Under development
Implementation Technology	Programming language: Python

5.1.3 Information Flow and Internal Architecture

In accordance with the information on section 5.1.1, the self-healing and self-correcting component will be able to detect when there is an issue with the TALON network and based of the information from the detected event some suggestions will be proposed for either resolving the issue (repair) or some action to be taken in order to avoid similar situations in the future. Finally, this component will be able to predict some event based on the real time data and historical from the TALON network and some actions will be suggested in order to avoid the disruption in the first place.

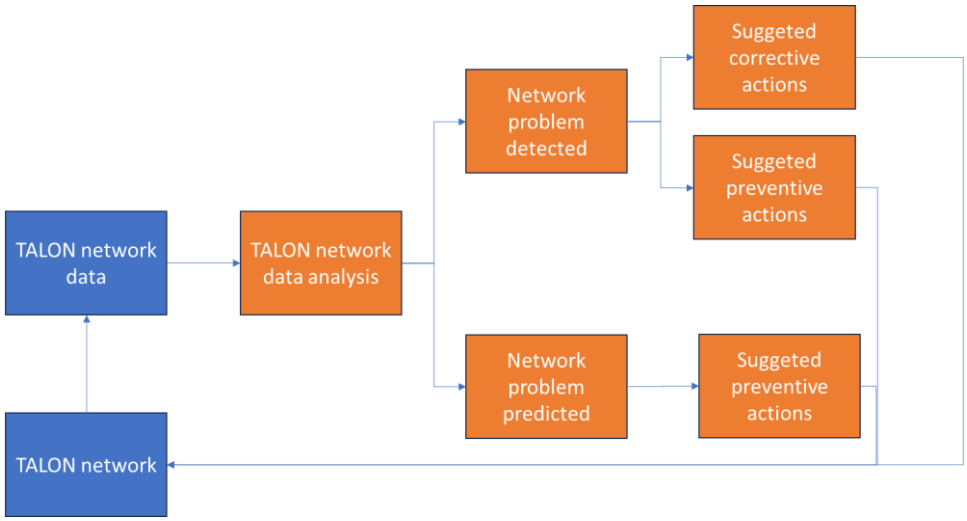

Figure 16 - Self-healing and Self-correcting component architecture.

5.1.4 Interaction with Other Components

The Self-healing and Self-correcting component interact with the following components.

Table 26 - Self-healing and Self-correcting interactions

Interacting Component	Description
TALON Dashboard	The Authentication and Authorisation component supports the mechanisms for secure interaction with the front-end web services comprising the Dashboard.
Data Monitoring Collection and Aggregation	The self-healing and self-correcting will retrieve data from the TALON network monitoring metrics and at the same time will make the necessary adjustments and corrections to the network to make sure high network efficiency.

5.2 AI Capabilities, Optimised and Transfer Learning

5.2.1 Business Logic

AI Capabilities and Transfer Learning component encapsulates three (3) key technologies of the TALON framework. In particular, the first technique that will be utilised in the context of TALON is Transfer Learning (TL), which aims to significantly reduce the learning latency by taking advantage of pre-trained AI models. Moreover, in order to further increase the performance of TALON's AI model without increasing the computational cost, the learning latency, or the effort cost related to data collection, Few-shot Learning will be used. Finally, with the main goal being the preservation of data privacy and security between different entities that participate in the training of AI models under the condition that their personal data will not be distributed, FL approaches will be implemented. Overall,

this component aims to optimise the trade-off between learning latency, and accuracy, while, simultaneously, contributing toward the privacy and the security of data.

As already stated, TL involves the utilisation of the knowledge gained from solving a particular problem to help solve a different, but related, problem. It is worth noting that TL has remarkable performance improvements, especially when working with limited data. In order to take full advantage of TL, an appropriate pre-trained model that has been trained on a large dataset related to the specification of the problem that we try to solve, should be initially selected. Next, after the proper preparation and pre-processing of data, the pre-trained model should be modified to adapt to the needs of the task by adding/removing certain layers, and modifying output layers, where needed. Then, the re-training of the model takes place using the new dataset and the modified model, while, in parallel, hyperparameter tuning is implemented to achieve optimal performance. Finally, after the training, the model is evaluated and tested and then deployed to perform the corresponding task.

The aforementioned sequence can be utilised in the implementation of Few-shot Learning. More precisely, while in real-world scenarios, it is costly, or time-consuming, to obtain a vast amount of data that traditional machine learning algorithms require, Few-shot Learning aims to provide high-performance models with only a small number of training examples. Since the training dataset is limited, the focus is on the selection of the model that needs to be as suitable as possible to perform the task that is required, and its generalisation ability.

The third technology that this component offers is FL. It is focused on collaboratively training an AI model across different nodes (i.e., different factories), without the need to exchange each node's personal data. More specifically, a server that orchestrates the whole training procedure of FL is responsible for the distribution and aggregation of the AI model, while each node individually trains the model and then sends it back to the server for aggregation. In doing so, the personal data of each node are kept private, and, simultaneously, the performance of the model remains high.

5.2.2 Functional Description

Table 27 - AI Capabilities and TL functional description

Component ID	TALON_AIC_AICOTL
Related Functional Requirement	FR13, FR14
Component Name	AI Capabilities and Transfer Learning
Description	AI Capabilities and Transfer Learning Component offers a set of technologies, targeting different aspects, either regarding the performance, or the privacy of data. In particular, Transfer Learning along with Few-shot and Federated Learning will be utilised.
Inputs & Pre-conditions	1. Data 2. AI model
Outputs & Post-conditions	1. Metrics regarding the task 2. Updated Models
API	Under Development
Implementation Technology	1. Python Programming Language 2. Flower – Federated Learning Framework 3. TF/Keras

	4. PyTorch

5.2.3 Information Flow and Internal Architecture

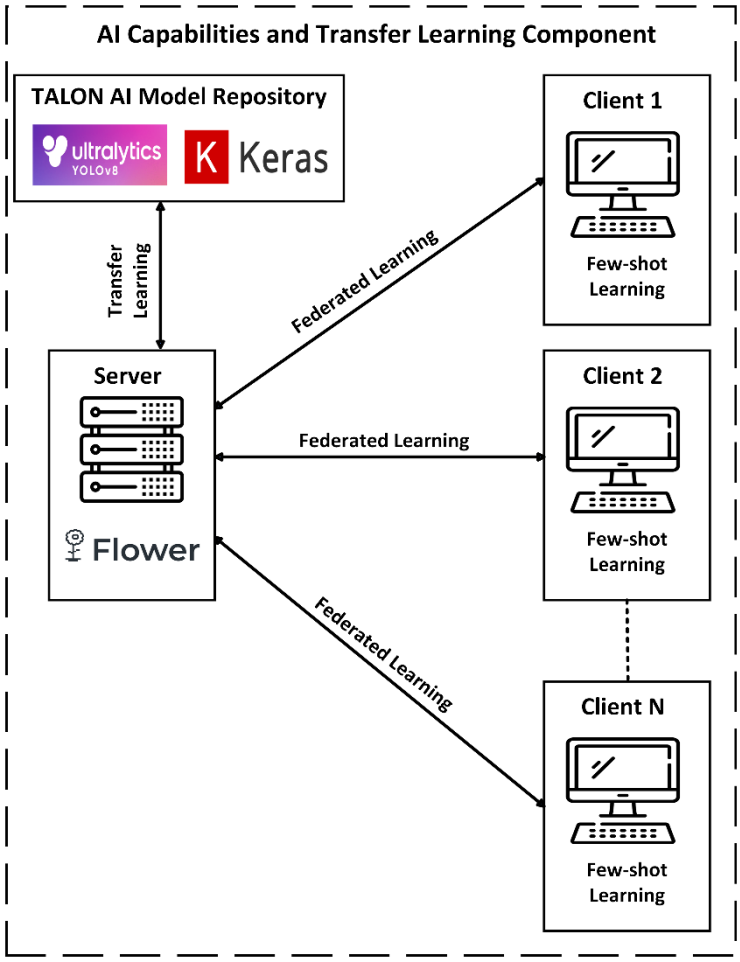

Figure 17 - AI Capabilities and Transfer Learning architecture

As it is shown in Figure 17, TALON's AI model Repository, which stores the trained AI models that will be utilised in the context of TALON, directly communicates with a central entity, like the server shown, in order to send the model and the respecting parameters (Transfer Learning). Then, the server initiates a Federated Learning process by distributing the AI model to the corresponding nodes. Next, each node, locally, trains the transmitted AI model using a few examples, utilising Few-shot Learning, and then sends back the trained model. The server aggregates the model of each node and sends back the updated one. This procedure continues for a specific number of training rounds, and then, the final model is transmitted back to the TALON AI Model Repository.

5.2.4 Interaction with Other Components

The AI Capabilities and Transfer Learning component interacts with the following components.

Table 28 - AI Capabilities and TL interactions

Interacting Component	Description

TALON Dashboard	The AI Capabilities and Transfer Learning component will communicate with TALON Dashboard to expose the corresponding results.
Polyglot Data Management (AI Model Repository)	The AI Capabilities and Transfer Learning component will communicate with TALON Polyglot Data Management which stores and retrieves the AI models to/from an AI Model Repository.

5.3 Data Operations

5.3.1 Business Logic

The data preprocessing component of TALON is responsible for the data preparation so that the data that originate from the TALON use cases is transformed into a format that is more easily and effectively leveraged by the ML-based models for training and for running inferences against them. The common 4 preprocessing steps undertaken by the TALON data preprocessing component are the following:

1) Anonymization and encryption: The raw data originating from the TALON use cases are either time-series or image data. These need to be properly encrypted and anonymized so that personal identifiable information existing in the data is removed.

2) Curation: The aim of the data cleaning step is to remove bad data, fill in missing data, format data (such as parsing dates), and ensure that the pre-processed data are in suitable for semantic enrichment. Moreover, it can deal with the problem of data augmentation, in those cases when training datasets are unbalanced.

3) Semantic enrichment: By using feature engineering, the role of the semantic-enrichment block is to select, manipulate, and transform the data that come as an output of the curation block into features that can be leveraged by the ML models.

4) Visualization and exploration: This final step of the TALON data preprocessing component involves the statistical visualization and exploration of the semantically enriched data so to identify and recognize trends and patterns. These patterns reveal valuable information about the data and give insights for building effective machine learning models and enhancing them for improving the model predictions.

5.3.2 Functional Description

Table 29 - Data Preprocessing functional description

Component ID	TALON_AIC_DO
Related Functional Requirement	FR15
Component Name	Data preprocessing
Description	The role of this component is to preprocess the raw data that originate from the TALON use cases. In particular, the data need to be: i) anonymized and encrypted; ii) cleaned; iii) semantically enriched; iv) visualized and explored.

Inputs & Pre-conditions	Use-case raw data and sets of actions/rules from the TALON XAI component.
Outputs & Post-conditions	Processed data as a training input to the distributed learning models in the E2C continuum and also to the TALON XAI component.
API	Report via Swagger OR OpenAPI will be made available **Curation:** swagger/#/curation/get_next_choice: this API allows the frontend to guide the user in choosing the correct curation algorithm based on XAI input. Semantic Alignment & Enrichment
Implementation Technology	Curation: A python backend is used in order to support data curation, in conjunction with libraries such as Albumentation and python-opencv (image transformations), pandas and scikit-learn (data processing).

5.3.3 Information Flow and Internal Architecture

The information flow of the data preprocessing component and its interaction with other components is depicted in Figure 18.

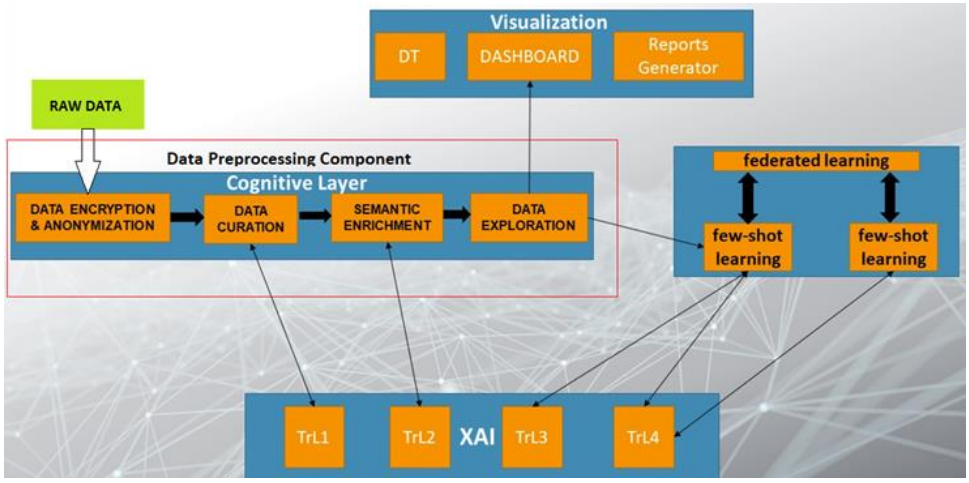

Figure 18 - Information flow of the TALON data preprocessing component and interaction with other components

Regarding the interaction with other components, during the data preprocessing phase the component gives input to and receives input from the XAI TALON component, as it is depicted in Figure 18. More specifically, the anonymized and encrypted data are forwarded to the XAI component that performs actions related to Trl 1(see Section 5.4), such as identifying imbalances in the data. Then, the XAI component suggests a set of actions to be taken by the engineers. The same holds with the curated data. These are again forwarded to the TALON XAI component that analyses them and suggests to the engineers a set of feature engineering actions that can be taken related, for instance, to dimensionality reduction, normalization, etc. This is related to Trl2 (see Section 5.4).

In addition, the pre-processed data are the final step of exploration are forwarded to the two-layered distributed federated learning framework, in which the model training is distributed across a number of edge devices that are selected as a solution of a well-defined optimization problem with constraints the energy autonomy of each device and its computational capabilities. Finally, the outputs of the TALON data preprocessing component will be communication to the TALON Visualization Dashboard that aims to visually explore the outputs of each TALON component (see Section 5.8).

Concerning the internal logical architecture of the blocks that comprise the TALON data pre-processing component, the architecture of the data encryption and anonymization component has been presented in Section 3.2.

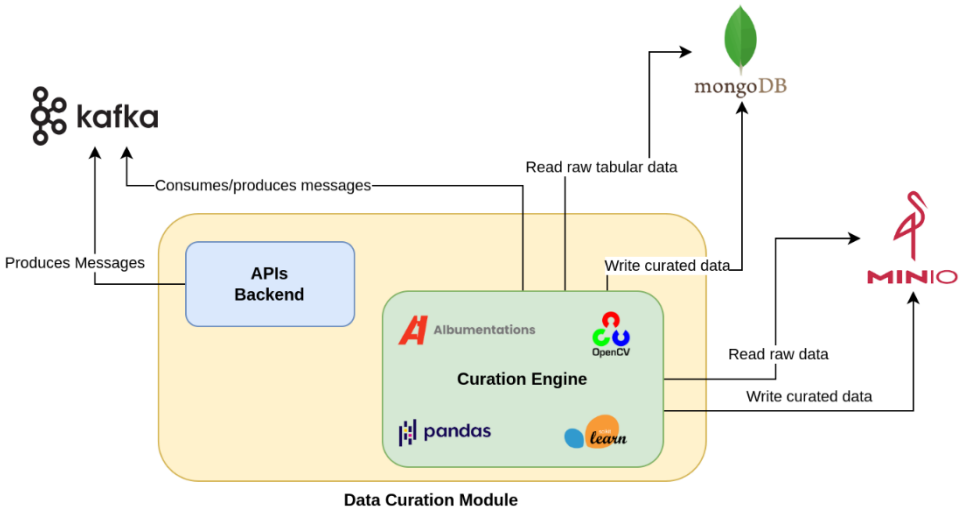

Figure 19 – Data curation architecture

As far as the data curation module is concerned, as shown in Figure 19, data curation module is composed of two main components:

- The APIs backend helps the user through the frontend to find the right curation algorithm to apply, after receiving the input from the XAI module. Moreover, it adds curation jobs to a Kafka queue.
- The Curation engine applies the algorithm to the data itself. It consumes messages from the Kafka queue and writes back results to the proper database. MongoDB is used for tabular datasets and time series, while Minio is used for filesystem-structured datasets (e.g., image datasets).

5.3.4 Interaction with Other Components

The Data Operations component interacts with the following components.

Table 30 - Data Preprocessing interactions

Interacting Component	Description
XAI and Monitoring	The TALON data preprocessing component gives input to and receives input from the XAI TALON component, as it is depicted. More specifically, the anonymized and encrypted data are forwarded to the XAI component that performs actions related to TrI 1, such as identifying imbalances in the data. Then, the XAI component suggests a set of actions to be taken by the engineers. The same holds with the curated data. These are again forwarded to the TALON XAI component that analyses them and suggests to the engineers a set of feature engineering actions that can be taken related, for instance, to dimensionality reduction, normalization, etc. This is related to TrI 2.
Optimised Learning	The output of the TALON data preprocessing component is forwarded as training to the TALON hybrid learning component that runs AI models based on FL and hybrid-shot learning across the TALON E2C continuum.
Visualisation Dashboard	The TALON data preprocessing component gives its output to the TALON visualization dashboard that visually explores the output of all the TALON components.

5.4 Digital Twin

5.4.1 Business Logic

The role of the Digital Twin is to provide an improved virtual (3D) representation of all the assets present in the working environment of an industry.

Digitized models of physical systems will be used to test and validate production strategies before taking them into practice. The DT will be used to design and select the resources of the assembly system corresponding to the safety requirements and its relationship with the rest of the system.

The virtual representations of the different physical devices will be accessible through APIs, creating a framework of consistent interoperability that allows the building of the digital twins in a more focused domain. An industrial model exchange standard is used for facilitating the integration of models with monitoring algorithms, protecting the model intellectual property, and making the framework independent from the modelling source.

Features:

- Provides the capability of building models and establishing the relationships between the inputs of the model and the collected data (contextualization).
- Provides the capability of loading, storing and updating individual models, e.g., videos and images collected from the surroundings and indoors of a factory, representing the different sections of the factory.
- Provides the capability of storing and visualizing the safety warnings in a more comprehensive manner and intuitive representation.

5.4.2 Functional Description

Table 31 - Digital Twin functional description

Component ID	TALON_AIC_DT
Related Functional Requirement	FR16
Component Name	Digital Twin
Description	The Digital Twin will provide a virtual representation of an industry. DT will be used to test and validate safety-related events during the execution of the actual scenarios
Inputs & Pre-conditions	Images/Videos/Video Streams
Outputs & Post-conditions	Improved Visualizations/3D renders
API	Following the architecture advances and the final definition of the use cases, OPEN APIs (OpenGL, DirectX) will be utilized and developed to support the data consumption of images (e.g., bmp files), videos (e.g., mkv files) and video streams (e.g., rtsp).
Implementation Technology	For the 3D renders (e.g., NeRF models) and in alignment with the scenarios defined, open-source python libraries will be used. For the detection of safety risk tasks, a drone and/or cameras will be used. Hence, for multi-object detection TensorFlow and OpenCV dependencies will be deployed using Python based libraries.

5.4.3 Information Flow and Internal Architecture

The information flow of the Digital Twin component is depicted in Figure 20.

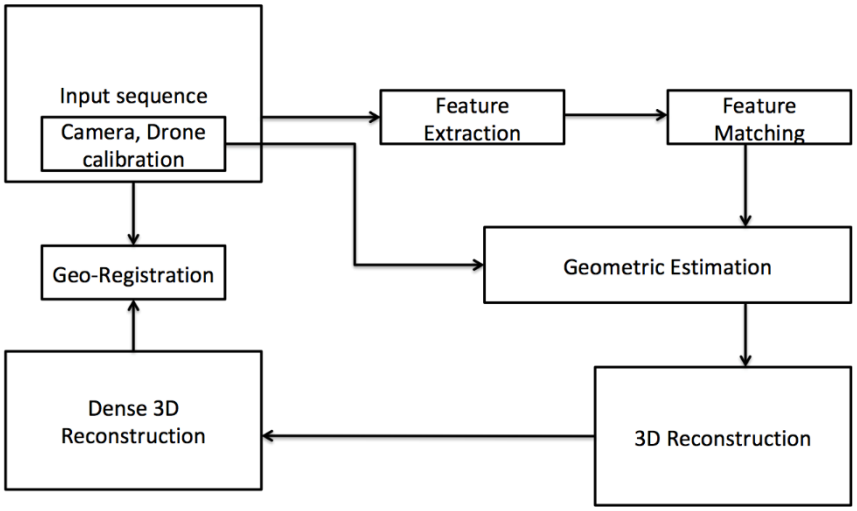

Figure 20 - Digital Twin workflow diagram.

DT is deployed as a docker container allowing to produce an improved, safety related representation environment of a manufacturing asset/plant based on image/videos data from the factory environment and obtain results that are visualized in different data visualization formats.

5.4.4 Interaction with Other Components

The Digital Twin component interacts with the following components.

Table 32 - Digital Twin interactions

Interacting Component	Description
TALON Dashboard	The Authentication and Authorisation component supports the mechanisms for secure interaction with the front-end web services comprising the Dashboard.

5.5 XAI and Monitoring

5.5.1 Business Logic

Enhancing the interpretability and understandability of artificial intelligence (AI) models and their decision-making process for human users is a key goal of Explainable Artificial Intelligence (XAI) [5]. Typical machine learning algorithms, such as deep neural networks, sometimes perform like black boxes, generating conclusions without revealing the reasoning behind them. These models can achieve high evaluation metrics, yet there are serious drawbacks due to their lack of transparency. XAI is significant in a number of cases. First of all, it tackles the crucial requirement for Accountability and Trustworthiness [6]. Users need to have faith in AI systems and comprehend the reasons behind the decisions that are made. XAI gives users the ability to verify the AI system's decisions by offering explanations that are comprehensible to humans. Second, a key component of XAI is interpretability. XAI improves model interpretability by providing clear justifications for AI predictions, particularly in fields where model choices have important real-world impacts. The interpretability of AI-driven judgments allows domain experts to understand, challenge, and potentially improve upon them.

Furthermore, XAI methods are essential for identifying and reducing biases in AI models. Developers can ensure fair and ethical AI deployments by identifying and correcting biases by bringing light on the features or data points driving predictions. What's more, XAI helps find problem areas in models. Knowing when and why AI models could anticipate things incorrectly enables prompt intervention and ongoing model enhancement, leading to more robust and dependable AI systems. Moreover, XAI is crucial for practitioners and researchers who want to develop AI. Transparent models help to build more potent and responsible AI technologies by offering insightful information about the potential and constraints of AI. Overall, XAI's significance lies in ensuring transparency, interpretability, and trustworthiness in AI systems. This fosters broader adoption of AI across various domains and facilitates effective human-AI collaboration.

In the context of the TALON project, Explainable Artificial Intelligence (XAI) will be applied in a non-sequential way across four distinct levels in different stages of TALON's life cycle, namely TrI1, TrI2, TrI3, and TrI4. Each level will address specific challenges related to both structured and unstructured data, using tabular data for timeseries and images, among others. At TrI1, XAI will be employed for tasks such as missing values detection in tabular data and verification of size, extensions, and identification of corrupted images for image data. This will help to enhance data quality and reliability. Moving to TrI2, XAI techniques will be used for outlier detection in tabular data and handling class imbalance issues in image datasets directories. By identifying outliers and addressing the class imbalance, this level aims to enhance the robustness and performance of the AI models applied in the subsequent stages. At TrI3, the emphasis will be on providing prediction explanations for the AI models used in both tabular and image data domains. XAI will enable us to gain insights into the factors influencing the model's predictions, making the decision-making process more interpretable. Finally, TrI4 will focus on generalising the insights gained from the previous levels and evaluating the performance of the AI models. XAI will be essential in understanding how well the models generalize to new data and how they perform in real-world scenarios. Evaluation metrics will be used to measure the effectiveness and reliability of the models.

Throughout the project life-cycle, the application of XAI in these the above levels will not only ensure interpretability and trustworthiness in the AI and ML models but also facilitate the effective handling of data challenges in the TALON project. By leveraging XAI at different stages, we aim to achieve robust, explainable, and high-performing AI solutions for a variety of data types.

5.5.2 Functional Description

Table 33 - XAI and Monitoring functional description.

Component ID	TALON_AIC_XAIM
Related Functional Requirement	FR17
Component Name	XAI and Monitoring
Description	XAI and Monitoring component is utilised in four (4) distinct levels, called Trust Levels (TrL) aiming provide transparency and trustworthiness of TALON framework, targeting different phases of the whole TALON's pipeline.
Inputs & Pre-conditions	1. Raw Data 2. Processed Data 3. AI models 4. Predictions

Outputs & Post-conditions	1. Visualisations 2. Insights & Findings
API	Under Development
Implementation Technology	1. Python

5.5.3 Information Flow and Internal Architecture

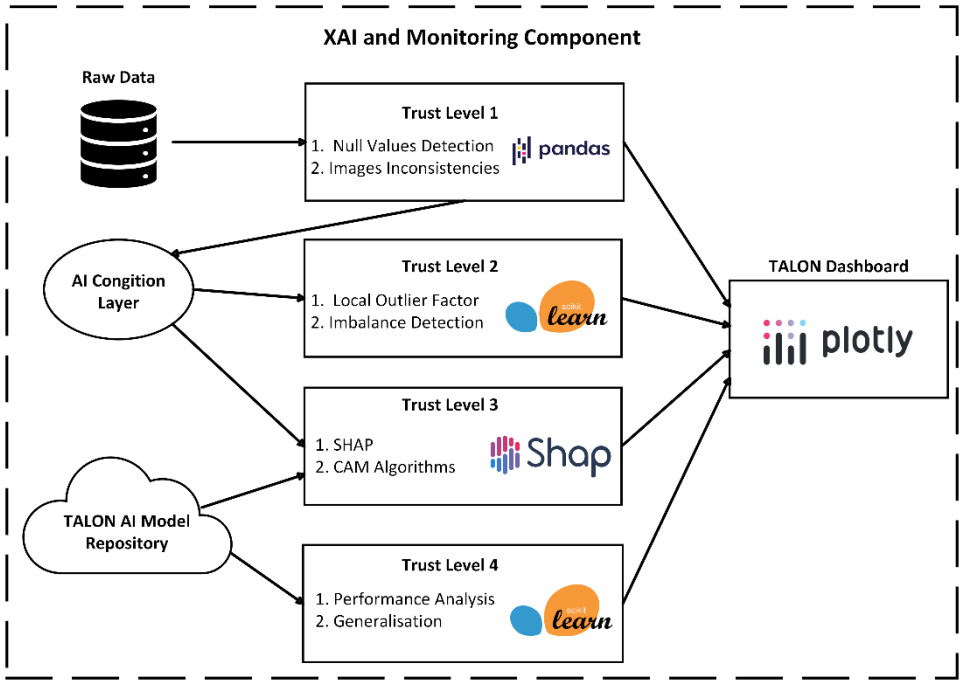

Figure 21 - XAI architecture

As already described and shown in Figure 21, each Trust Level (TrL) targets a different phase of the TALON's framework pipeline. Starting from TrL1, it takes as input the raw data in order to validate and check their source reliability and report any inconsistencies in the data. Next, TrL2 directly interacts with the Cognition Layer to investigate various aspects of the processed data, while TrL3 and TrL4 communicate with the TALON AI Model Repository to provide explanations regarding the predictions that these models made, and their generalisation ability along with the corresponding performance metrics. Finally, it is important to note that the insights of each TrL are communicated to the TALON Dashboard in order to be visualised, utilising the most appropriate visualisation plot.

5.5.4 Interaction with Other Components

The XAI and Monitoring component interacts with the following components.

Table 34 - XAI and Monitoring interactions

Interacting Component	Description
TALON Dashboard	The XAI and Monitoring component will interact with the TALON Dashboard to visualise each TrLs insights.

5.6 Polyglot Data Management

5.6.1 Business Logic

The role of the Polyglot Data Management is to manage data persistency and real-time services interaction. The TALON vision raises several challenges regarding data persistency, both in terms of sending/receiving, storing and retrieving data. Its main purpose is the combination of existing, collected, and business datasets, as well as data collected for the AI Orchestrator. The data may have different schemas, while it can be structured, semi-structured or fully unstructured.

Apart from that, each service provider or TALON pilot partner might require authentication, data protection and access rights, according to the level of authorisation of the user, in order for the latter to be able to monitor probes, collect cyber-related information for anomaly detection and efficiently manage and store it. As a result, the combination of these diverse data sources is a challenging task.

TALON will employ a Polyglot Data Management component for storing various aspects of information that is being generated and consumed within the lifecycle of the various components complimented by a message queueing system. The primary flavour of the polyglot storage is the transactional storage (with atomicity, consistency, isolation, and durability (ACID) guarantees). This type of storage will be used to store existing data from the pilot partners such as sensorial data and metadata, monitoring (i.e., from network to application level) and performance-related data. It should be clarified that the polyglot data management consists of three (3) types of databases; one relational (e.g., PostgreSQL[9]), one object store (e.g., MinIO[10]) and one NoSQL (e.g., MongoDB[11]). The relational database is used in order to store fully structured data that change rarely (e.g., master data of the system, credentials, business partners data, etc.) while the NoSQL is used in order to store semi-structured data that change frequently (e.g., data that are changing through the data lifecycle management from data curation to semantic enrichment). The object store will be used to manage unstructured data such as photos, videos, etc.

The message queueing system (e.g., Kafka[12]) is used to decouple the communication of the TALON components and more specifically to eliminate any blocking communication that may be required. Elimination of blocking communication is a prerequisite for the creation of a scalable system. Due to the fact that REST APIs will be widely used for several communications and since the HTTP request/response protocol is best used in cases that call for a synchronous request/reply, services in TALON System must be designed to expect an immediate response. This means that asynchronous communication between services also has to be supported and it will improve the fault tolerance and resiliency within the entire system. Towards this, message brokering and queueing services and mechanisms need to be built to enable the matching and rooting communications, the support of different transport protocols (if needed), the proper transformation of message formats among all communicated services / pods, and the identification and distribution of events for the various disparate sources.

On top of the Polyglot Data Management component, a distributed search and analytics engine (e.g., Elastic[13], InfluxDB[14]) for all types of data collected for the AI Orchestrator is envisioned to support the reporting and analysis tasks.

[9] https://www.postgresql.org/
[10] https://min.io/
[11] https://www.mongodb.com/
[12] https://kafka.apache.org/
[13] https://www.elastic.co/
[14] https://www.influxdata.com/

5.6.2 Functional Description

Table 35 - Polyglot Data Management functional description

Component ID	TALON_AIC_PDM
Related Functional Requirement	FR18
Component Name	Polyglot Data Management
Description	A relational database stores and provides access to data points that are related to one another, according to a data model which is an intuitive and straightforward way of representing data in tables. A NoSQL database provides a mechanism for storage and retrieval of data that is modelled in means other than the tabular relations used in relational databases. Search engine allows search and analysis of huge volumes of data quickly and in near real-time and give back answers in milliseconds. It is able to achieve fast search responses because instead of searching the text directly, it searches an index. It uses a structure based on documents instead of tables and schemas and comes with extensive REST APIs for storing and searching the data. The message queue serves as a form of asynchronous service-to-service communication used in serverless and microservices architectures, where messages are stored on the queue until they are processed and deleted.
Inputs & Pre-conditions	Connection details configured in Python, Java or more (i.e., connections details should be specified including server IP and port (i.e., 5432, 27017, 9343 with TLS/SSL, 9243, topics where services and applications can be connected to produce/consume data)).
Outputs & Post-conditions	Successful connection/read/write to the respective database system.
API	https://www.postgresql.org/docs/15/index.html https://www.mongodb.com/docs/

	https://min.io/docs/minio/linux/index.html https://kafka.apache.org/documentation/ https://docs.influxdata.com/influxdb/v1.8/guides/ https://www.elastic.co/guide/index.html
Implementation Technology	All programming languages can be supported. The frameworks which will be used are: • PostgreSQL • MongoDB • MinIO • Kafka • InfluxDB • Elastic

5.6.3 Information Flow and Internal Architecture

In order to deal with all the aforementioned challenges, TALON will benefit from the use of PostgreSQL, MongoDB, MinIO, Kafka, InfuxDB and Elastic platforms and the developments that will be built on top of them in the context of the project. As a result, these platforms will be the basis for the Polyglot Data Management component as they bring a number of capabilities that are related to the needs of the project. The technologies which will be used are as follows:

- PostgreSQL is a powerful, open-source object-relational database system with a strong reputation for reliability, feature robustness, and performance.
- MongoDB is a source-available cross-platform document-oriented database system. Classified as a NoSQL database system, MongoDB uses JSON-like documents with optional schemas.
- MinIO is a high-performance object storage which can handle unstructured data such as photos, videos, log files, backups, and container images with a current maximum supported object size of 50TB.
- Kafka is an open-source distributed event streaming platform for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.
- InfuxDB is a high-speed read and write time-series database which is used for storage and retrieval of time series data in fields such as operations monitoring, application metrics, Internet of Things sensor data, and real-time analytics.
- Elastic is a distributed, free and open search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured.

The high-level architecture related with the Polyglot Data Management component is depicted in Figure 22.

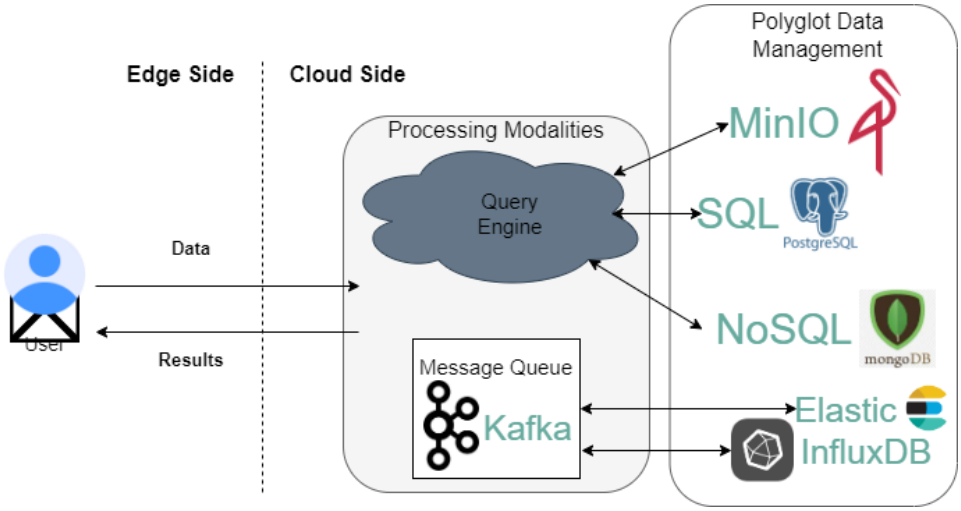

Figure 22 - Polyglot Data Management component architecture

5.6.4 Interaction with Other Components

The Polyglot Data Management component interacts with the following components.

Table 36 - Polyglot Data Management interactions

Interacting Component	Description
All TALON Components & Third Party services	The Polyglot Data Management indexes, stores and provides all the data required for reporting, analysis and visualisation purposes by the different TALON Components as well as Third Party Services

5.7 Visualisation Dashboard

5.7.1 Business Logic

The role of the Visualisation Dashboard is to collect and present in a user-friendly manner the reports or the outputs provided by each TALON component. The visualization items are delivered to the user via the TALON Dashboard which is the interface the users will use to interact with the TALON System. The TALON Dashboard is where the users will land once they sign in TALON System through the Authentication and Authorisation component. It will allow the users to:

- Monitor the E2C metrics, filter by identifier and view/edit the smart policies of the orchestrator.
- Visually explore the results, get reports and notifications.
- Get the results of the energy and distributed intelligence / network analytics.
- Get the results of the resources analytics.

5.7.2 Functional Description

Table 37 - Visualisation Dashboard functional description

Component ID	TALON_AIC_VD
Related Functional Requirement	FR19

Component Name	Visualisation Dashboard
Description	A collection of numerical, textual and graphical information depicted in visual forms (e.g., charts for bars, lines, etc., heatmaps, etc.) derived from each TALON component's performed functionality.
Inputs & Pre-conditions	Analytics reports in the form of JSON
Outputs & Post-conditions	Visual elements and charts of the reports and the results
API	http://talon-ui.euprojects.net/opendoc/
Implementation Technology	Javascript Angular JS framework

5.7.3 Information Flow and Internal Architecture

Regarding the Visualisation Dashboard, no specific internal architecture is envisioned because this layer will consolidate within a front-end web application, the exposed services and APIs supported by the other TALON Components. It will basically interact with the Polyglot Data Management either to initiate services (e.g., Policies Editor, etc.) or to get the results produced by the services (e.g., reports produced by the Anomaly Detection component, Resource or Energy optimisations, etc.).

5.7.4 Interaction with Other Components

The Visualisation Dashboard component interacts with the following components.

Table 38 - Visualisation Dashboard interactions

Interacting Component	Description
Polyglot Data Management	The Visualisation Dashboard interacts with the Time-Series, Search Engine and the Relational database of the Polyglot Data Management component.
Authentication and Authorisation	The Visualisation Dashboard interacts with the Authentication and Authorisation component regarding the User Authorization and Authentication.

6 Requirements Traceability Matrix

We provide below the Requirements Traceability Matrix (RTM). The matrix links the stakeholder requirements as defined in D2.1-Use Case, KPIs, Requirements, Specification, Slices & Technology Enablers Definition Report with the TALON functional requirements and technical components presented in this deliverable. We will continuously monitor RTM to assess the progress of each individual technical component as the project evolves.

Table 39 - Requirements Traceability Matrix

Stakeholder Requirement	**Use Case Supported**	**Functional Requirement**	**Component Name**	*Component Description*
StRS01	UC1; UC2; UC3; UC4	FR01	Authentication and Authorisation	The authentication and authorisation functionality will control the access to the E2C TALON resources and ensure the confidentiality, integrity, and availability of sensitive information.
StRS02, StRS20, StRS24	UC1; UC3; UC4	FR08	Resource Allocation and Deployment	The optimal resource allocation and deployment will efficiently distribute resources to achieve the best possible outcome or maximize a certain objective. It involves making decisions on how to allocate resources such as time, idle slots and computing nodes in a manner that optimises data and services placement, response time, and energy efficiency.
StRS03	UC1; UC3; UC4	FR05	Application Lifecycle Manager	Application Lifecycle Manager will graphically describe application topology, denote intercommunication and dependencies among the applications' services and sub-components. Structured YAML Ain't Markup Language (YAML) or JSON files will be used for the configurations.
StRS04, StRS05, StRS06, StRS07	UC1	FR07	AI Swarm Orchestration	Swarm orchestration will manage and coordinate a swarm of autonomous entities or agents to achieve a common goal. In swarm orchestration, multiple individual agents will work together in a decentralised manner, leveraging local interactions and simple rules to exhibit emergent behaviours and accomplish complex tasks, such as obstacle avoidance, trajectory, and energy planning.
StRS08	UC2	FR04	Anomaly Detection	Anomaly detection will identify patterns, suspicious activities, errors, or changes in the system that deviate significantly from the expected behaviour within a network. It will automatically detect any

				anomalies that may require further investigation or immediate action.
StRS09	UC2	FR15	Data Operations	Data operations including collection, cleaning, enrichment, and alignment, as well as querying for exploratory data analysis will be performed on the collected data to produce reports and ensure that the tools are working within their optimum range.
StRS10, StRS21	UC2; UC3	FR13, FR14	AI Capabilities, Optimised and Transfer Learning	The AI capabilities for scene analysis and safety level detection (PPE, fire) will be trained using transfer learning to leverage knowledge learnt from one task or domain to improve performance on a different but related task or domain. Also, Optimised Learning will detect and report on breakage and damages of the cutting tools; support meta-learning to learn how to quickly adapt to new tasks or categories using a limited amount of labelled data; data augmentation to increase the diversity of the supporting dataset; and generalisation on top of a few labelled examples to overcome the data unavailability.
StRS11, StRS13	UC2	FR12	Self-healing and Self-correcting	The component will develop and train AI models to autonomously detect, diagnose, and repair damage or faults without external intervention. The AI algorithms will learn over the collected and curated data to facilitate the prompt diagnosis and evaluation of breakage incidents in the manufacturing contexts. Also, different AI models will be trained to benchmark, compare, and select the most suitable model per case.
StRS12	UC1, UC2, UC3, UC4	FR18	Polyglot Data Management	The Polyglot Data Management will support the underlying database and distributed storage systems of the TALON project and the back-end processes to manage data, from its creation or acquisition to its eventual storage and querying.
StRS14	UC2; UC3; UC4	FR19	Visualisation Dashboard	The visualisation dashboard will support a graphical user interface that displays visual representations of data, metrics, and key performance indicators (KPIs) of the TALON project in a consolidated and easily understandable format.

StRS15	UC2	FR17	XAI and Monitoring	The TALON System will provide understandable explanations regarding optical link faults, imbalanced data and data that need curation and cleaning. Through a stepwise approach, it should support transparency, trustworthiness and interpretability of the data and the results of black-box AI models.
StRS16	UC2; UC3; UC4	FR16	Digital Twin	Digital Twin will simulate and optimize manufacturing processes, monitor equipment performance, monitor working condition safety measures via augmented images and predict maintenance needs.
StRS17	UC3; UC4	FR02	Data Anonymisation	Data anonymisation functionality will remove and alter personally identifiable information (PII) from data sets in such a way that the data can no longer be linked to specific individuals. The goal of data anonymisation will be to protect privacy while still allowing for analysis, research, or other legitimate uses of the data. In TALON, facial, textual, and numerical anonymisation will be supported.
StRS18	UC3	FR06	NG-SDN and Distributed Intelligence	The Distributed Intelligence will abstract the control plane and move it to a centralised controller, which communicates with the network devices. This separation will allow for programmability, flexibility, and easier management of the network.
StRS19	UC3; UC4	FR03	DLTs for Securing AI/ML models weights	A distributed ledger, such as blockchain, will be employed to secure AI/ML model weights for (i) data integrity (i.e., the weights of AI/ML models can be stored on a distributed ledger, ensuring their integrity and preventing unauthorised modifications); (ii) version control and provenance (i.e., the ledger can maintain a transparent and immutable history of model weight updates, allowing for version control and easy tracking of changes); and consensus (i.e., agreement upon the correctness of model weight updates in federated machine learning settings).
StRS22	UC1, UC3	FR09	Smart Policy Manager	The Smart Policy Manager employs a metrics definition and monitoring for Key Performance Indicators (KPIs) linked with SLOs within the platform. This process will initially provide a schema which will hold every metric

				necessary to describe specific KPIs and then develop a method to keep track of these metrics, possibly in real time, for the system to self-improve and self-evaluate the performance of the platform.
StRS23, StRS25	UC1; UC3	FR11	AI Model Training, Adaptations and SLOs	The TALON System will provide the capability to train AI models and support smart and adaptive strategies by means of policies for the metrics collected by FR09 to meet the respective business objectives, KPIs and SLOs.
StRS26	UC1; UC3; UC4	FR10	Data Monitoring, Collection and Aggregation	The TALON System will periodically pull monitoring data from all its assigned agents. The orchestration system will periodically pull aggregated monitoring data from the storage system of the TALON Orchestrator and the Edge devices for all deployed applications. FR10 will give input to FR11 to train and enforce smart orchestration strategies.

7 Partners assets and background technologies

Here, the assets provided to the project and made available by the different partners will be listed in a summary table in which each one will have to transcribe the info related to it.

Table 40 - Partner Assets

Background Technology	**TALON Component**	**IPR Owner**	**TALON Start ⟶ End TRL**
SDN Lab: an experimentation software platform based on SDN, for validation of the new autonomous network-slicing algorithms and their performance evaluation in integrated systems	NG-SDN and Distributed Intelligence	UL	7 ⟶ 8
IO Toolkit Generator (S-040-7-2020): is a low code software development tool aimed to create affordable data integration microservices to connect manufacturing devices	Self-healing and Self-correcting	UPV	7 ⟶ 8
Visual-aided Intrusion Detection System (SiVi©): uses state of the art supervised and unsupervised ML algorithms in order to classify the incoming traffic and depicts the status of the network in a simple, but informative way.	Anomaly Detection	SID	5 ⟶ 8
AI4CI Toolkit: is the toolkit destined for critical infrastructure applications. AI4CI is a powerful AI suite of customisable frameworks, tools and processes designed to help you effectively utilize AI's potential.	AI Capabilities, Optimised and Transfer Learning	SID	2 ⟶ 5
CNC Machining Assets	Data Monitoring, Collection and Aggregation (at the end user application level)	FACT	9
AiRFLOW: A horizontal platform thar enables full UAV fleet management control and integration among various devices.	AI Swarm Orchestration	PROBO	7 ⟶ 8
Mission control centre (MCC) is an integrated environment that enables professional pilots to fly drones with safety and stability. MCC is fault tolerant and	AI Swarm Orchestration	PROBO	7 ⟶ 8

enables parallel control in case of catastrophic failures and is one of the few enterprise-ready solutions in the market.			
Human detection and identification toolkit	AI Capabilities, Optimised and Transfer Learning	CERTH	5 ⟶ 7
Human pose tracking software	AI Capabilities, Optimised and Transfer Learning	CERTH	5 ⟶ 7
Human activity recognition and behaviour monitoring toolkit	Digital Twin	CERTH	5 ⟶ 7
8BELLS Analytic Toolset Forecasting Demand / Sales. It estimates a potential demand so as to be able to make accurate decisions about pricing, business growth and market potential.	Smart Policy Manager (partially)	8BELLS	3 ⟶ 5

8 Conclusions

The D3.1 "Architecture & Platform Design Blueprint" document is a significant milestone in the TALON project, presenting the blueprint for a powerful architecture determined to revolutionize Industry 4.0/5.0. Building upon the foundation laid in previous deliverable D2.1 "Use Case, KPIs, Requirements, Specification, Slices & Technology Enablers Definition Report," we have here crafted an architecture that addresses diverse industrial use cases and challenges.

At the heart of our design there's orchestration of computational resources, AI models, and data to optimize system effectiveness and performance. Dedication towards energy efficiency drives us to establish greener AI networks, which support global sustainability efforts.

The high-level architecture is introduced in Section 2, which gives a comprehensive overview of the essential elements and TALON actors discovered from the input of preceding deliverable.

Sections 3, 4, and 5 delve deeper into the essential components of access and security, AI-fuelled orchestration, and AI cognition. Each component is designed with business logic, functionalities, and seamless interaction with others, enabling efficient interoperability within the TALON ecosystem.

The Requirements Traceability Matrix (Section 6) demonstrates how requirements from stakeholders have been successfully included, ensuring that our design perfectly satisfies their needs and aspirations, providing transformative solutions.

Section 7 enlists partner assets, exploiting their collective knowledge and expertise in utilized technologies.

Finally, the D3.1 “Architecture & Platform Design Blueprint" embodies our commitment to advancing AI innovation, orchestration, and sustainability in Industry 4.0/5.0. This microservice-based infrastructure represents a dynamic, efficient, and impactful system. As we move forward, the realization of our vision, expressed in this document, will give rise to a platform demonstrated in deliverable D3.6, solidifying TALON's position in the industry.

References

[1] M. R. Rahimi, et al., “Mobile Cloud Computing: A Survey, State of Art and Future Directions,” Mobile Networks and Applications, 2014.

[2] S. Wang, et al., “A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications,” IEEE Access, 2017.

[3] T. Taleb, et al., “On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration,” IEEE Com. Surveys & Tutorials, 2017.

[4] C. Wang, et al., “Mobility-Assisted Opportunistic Computation Offloading,” IEEE Communications Letters, 2014.

[5] Ribeiro, M. T., Singh, S., & Guestrin, C. "Why should I trust you?" Explaining the predictions of any classifier, 2016. doi: https://doi.org/10.48550/arXiv.1602.04938

[6] Doshi-Velez, F., & Kim, B. Towards a rigorous science of interpretable machine learning, 2017. doi: https://doi.org/10.48550/arXiv.1702.08608

[7] R. Pinto, G. Gonçalves, J. Delsing and E. Tovar, “Enabling data-driven anomaly detection by design in cyber-physical production systems,” Cybersecurity, 2022.

[8] J. Raynor, T. Crnovrsanin, S. Di Bartolomeo, L. South, D. Saffo and C. Dunne, “The State of the Art in BGP Visualization Tools: A Mapping of Visualization Techniques to Cyberattack Types,” IEEE Transactions on Visualization and Computer Graphics, vol. 29, no. 1, pp. 1059 - 1069, 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