AI-Enabled Data Lifecycles Optimization and Data Spaces Integration for Increased Efficiency and Interoperability

Project acronym:	PLIADES
Project name:	AI-Enabled Data Lifecycles Optimization and Data Spaces Integration for Increased Efficiency and Interoperability
Grant Agreement No:	101135988
Call:	HORIZON-CL4-2023-DATA-01
Topic	HORIZON-CL4-2023-DATA-01-02
Type of action:	HORIZON Research and Innovation Actions
Start of project:	01/01/2024

Deliverable 2.3

System Technical Specifications and PLIADES framework Architecture

Work Package:	**WP2 – User Requirements and Specifications**
Task:	**T2.3**
Lead Beneficiary:	**CERTH**
Due Date:	**31/12/2024**
Submission Date:	**23/12/2024**
Deliverable Status:	**Final**
Deliverable Type:	**Report**
Dissemination Level:	**Public**

Authors

Surname	First Name	Beneficiary
Mariolis	Ioannis	CERTH
Tsiakas	Kosmas	CERTH
Nasoulis	Christos	CERTH
Tsamis	Georgios	CERTH
Alexiou	Dimitrios	CERTH
Giouvanis	Georgios	CERTH
Alonso Muñoz	Asier	TECNALIA
Ashneet	Khandpur Singh	EURECAT
Nandi	Arijit	EURECAT
Papaspyropoulou	Maria	ATLANTIS
Ortega Valdivieso	Juan Diego	VICOM
Mastandrea	Guiseppe	E@W
Rodriguez	Silvia	INNOVALIA
-	-	BOR, THI, UC3M, MU-EPS, KU Leuven, CEIT, IDSA, TNO, CICBiogune, BasqueCCAM, AVL, HIT, Libattion, Sipbb[1]

Reviewers

Surname	First Name	Beneficiary
Raposo	Marcio	DENN
Matowicki	Michal	CVUT

Version History

Version	Date	Modifications made by
0.1	22/10/2024	CERTH
0.2	06/12/2024	CERTH
0.3	10/12/2024	CERTH
0.4	16/12/2024	CERTH
1.0	23/12/2024	CERTH

[1] The listed partners provided their input indirectly through circulated templates processed by CERTH that authored the corresponding sections of the deliverable based on the consolidated feedback.

Disclaimer

Executive Summary

Deliverable D2.3, titled "System Technical Specifications and PLIADES Framework Architecture", provides a detailed overview of the architecture and technical specifications of the PLIADES platform. Building on the user requirements and use cases outlined in Deliverable D2.2 , "User Requirements and Use Cases", this document translates these into a high-level architecture that serves as the foundation for system implementation. The deliverable encompasses the methodologies for deriving technical specifications, the conceptual and functional architecture and the system's implementation framework.

The PLIADES architecture is structured into modular components that address the project's objectives of optimizing data lifecycles and enhancing interoperability within data spaces. By employing a hierarchical methodology, the deliverable progressively details the system nodes, their dependencies, and corresponding functionalities, enabling independent node development and seamless integration.

In alignment with the prioritized user requirements and PLIADES use cases, the deliverable defines the technical specifications, including performance, hardware and software requirements, as well as, the mapping of functional requirements. This structured mapping ensures that each component of the system is tailored to meet the specific needs of end-users while supporting the overall project goals.

Additionally, this deliverable introduces the development and deployment views of the PLIADES system. The development view outlines guidelines for module development, software configuration, and partner responsibilities, while the deployment view provides a comprehensive look at hardware and software integration.

The deliverable presents an implementation roadmap, detailing a timeline for addressing functional requirements during the project. This roadmap ensures consistency between the conceptual design and practical deployment, driving the PLIADES project's mission to advance AI-enabled data lifecycle management and interoperability.

Future updates and advancements during development and deployment of the PLIADES solutions will be documented in future deliverables in WP7 "Deployment in Diverse Use Cases for Data Spaces Integration and Cross Domain Validation".

Table of Contents

List of Figures

List of Tables

List of Terms and Definitions

Table 1 Definitions

Abbreviation	Definition
ADAS/AD	Advanced Driver Assistance & Autonomous Driving Systems
AI	Artificial Intelligence
API	Application Programming Interface
BESS	Battery Energy Storage Systems
CCAM	Connected Cooperative and Automated Mobility
CPU	Central Processing Unit
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
IDS	International Data Spaces
LLM	Large Language Models
SFR	System Functional Requirement
UC	Use Case
UR	User Requirement

1 Introduction

1.1 Scope of the deliverable

The purpose of this deliverable (D2.3 "System Technical Specifications and PLIADES Framework Architecture") is to outline the overall architecture of the PLIADES platform and provide a comprehensive blueprint to guide technical partners in implementing the proposed design. The primary objective is to translate the initial PLIADES concept into high-level nodes, each further divided into specific functional components. These functional components serve as the foundational software elements (nodes) of the PLIADES platform, mapped to implementation tasks on a unified software platform agreed upon by all partners. These implementations are validated through realistic use case scenarios, which are detailed in deliverable D2.2 "User Requirements and Use Cases". The current deliverable also presents the conceptual architecture's design methodology, defines the core system functionalities, conducts a preliminary technical analysis for each use case, and provides an overview of the development and deployment framework.

1.2 Relation to other Activities and Deliverables

The current report describes the results from Task T2.3 "Design and architecture of the platform and data integration strategies" of the PLIADES project, building upon other WP2 tasks, particularly Task T2.2, "Consolidation of User and System Requirements". Task T2.2 lays the groundwork by defining essential requirements for PLIADES, supporting D2.2, "User Requirements and Use Cases," which will be submitted alongside D2.3. D2.2 analyzes use cases to establish specific user needs, while D2.3 expands upon these with detailed functional and non-functional specifications, providing a framework for architectural development in work packages WP3, WP4, WP5, and WP6.

1.3 Structure of the deliverable

The deliverable is structured as reported below:

Section 1 introduces this report, by outlining the scope of the document, its relation to other activities and deliverables, and an overview of the structure of the deliverable.

Section 2 explains the methodology used for defining the PLIADES architecture, outlining fundamental principles for developing the conceptual architecture and specifies the approach for producing detailed software node definitions. Additionally, it describes the process for establishing the platform's technical specifications.

Section 3 outlines the conceptual architecture of the PLIADES system, detailing primary software nodes and dependencies.

Section 4 provides a functional analysis of the PLIADES system, mapping functional system requirements to identified use cases and sub-use cases of the system.

Section 5 elaborates on the functional view on the introduced conceptual architecture, specifying functional modules, responsibilities, and interactions among components.

Section 6 presents technical specification of the PLIADES platform, covering hardware and software requirements and detailing the alignment from use cases to specifications.

Section 7 presents the integration of the PLIADES architecture with the existing IDSA architecture.

Section 8 presents the implementation plan, including a timeline for addressing functional requirements throughout the project. It defines the development view, with architecture support for development, partner responsibilities, and software constraints. It also introduces the deployment view, specifying software node requirements, hardware, and partner-provided tools fulfilling sub-use case demands.

Section 9 concludes the document, summarizing key insights and providing an overall assessment of the deliverable's contribution to the PLIADES project.

2 Architecture and Technical Specifications Derivation Methodology

2.1 System Architecture Concepts and Design Principles

The architectural design process for PLIADES platform was conducted according to guidelines from both the “Software Systems Architecture” handbook [3] and from the “Fundamentals of Software Architecture” [4]. The PLIADES architecture comprises of multiple interconnected system nodes that include software components, externally visible properties, and defined relationships among them. The architectural design process aims to outline each platform component and its interactions to support overall functionality across the entire system. Key guiding principles for the architecture include:

- **Open and Modular Architecture:** The PLIADES platform is designed to be both modular and open, enabling flexible use of defined modules (nodes) by end-users and system integrators. This modular approach supports easy adaptation and integration of components of new or updated components, accommodating future system expansions.
- **Technology Independence:** The architecture promotes technology-agnostic solutions, prioritizing standardized and widely accessible technologies, both open-source and commercial, to maximize interoperability, adaptability, and long-term scalability.

To establish the software architecture for PLIADES, a hierarchical approach was implemented, as illustrated in Figure 1. Particularly, this architecture applies a hierarchical approach, beginning with PLIADES high-level goals to guide the preliminary analysis of essential architectural elements, presented in an abstract form. The **Conceptual Architecture Analysis** of the platform defines the core hardware and software components that form the basis of PLIADES functionality, establishing a clear framework for the system’s capabilities and interactions.

The final step involves the **Functional View**, which specifies each node’s internal structure and interactions, detailing component responsibilities and interconnections required for seamless platform integration. The Functional View also identifies the physical interconnections between hardware components and subsystems, facilitating coordinated operation across the entire system of PLIADES.

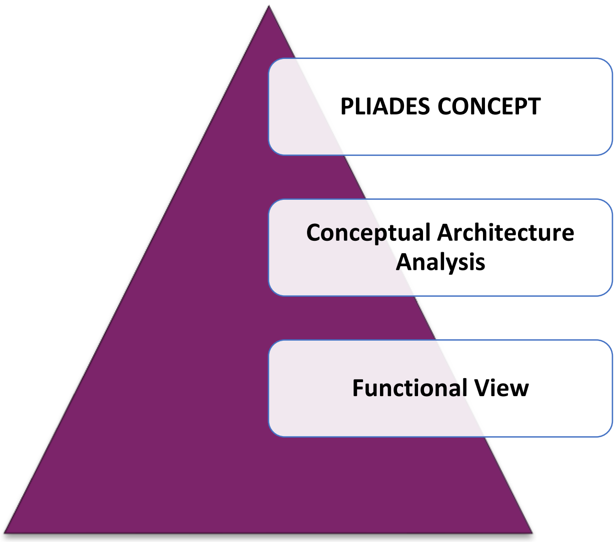

Figure 1 Hierarchical analysis of the PLIADES architecture

2.2 Architectural Development Process

This section explains the methodology used for designing the architectural components of the PLIADES platform, focusing on their functionalities and interactions. The development process is structured in stages, beginning with the conceptual architecture, and progressing to the high-level functional view. The key steps in this process are as follows:

- **Step 1 – User Requirements and Use Cases**: User requirements use cases, and sub-use cases will be outlined as part of Task T2.2 – "Consolidation of User and System Requirements", and will be detailed in Deliverable D2.2 – "User Requirements and Use Cases". This foundational analysis is crucial for structuring the PLIADES architecture, ensuring that all design components align with the specific needs identified across various project scenarios and end-users' requirements.
- **Step 2 – Technical Specifications:** Technical specifications for the PLIADES platform are defined based on user requirements, use cases and sub-use cases. This includes a conceptual breakdown of required hardware and software components, integrating performance requirements to shape these structures. Functional system requirements serve as a connection between user-centered needs and technical specifications, ensuring alignment across system design elements.
- **Step 3 – Functional View Analysis and Node Specification:** Specific tasks within each sub-use case are accomplished through a sequence of system functionalities, executed by specific software nodes. The functional view of the PLIADES platform provides a thorough analysis of these nodes, detailing each component's functionalities, dependencies, operational requirements, and indicative performance requirements. This in-depth perspective clarifies the role of each node, enabling effective integration and alignment with overall platform objectives.
- **Step 4 – System Implementation:** Building upon the functional view of PLIADES, a comprehensive plan for software node development is created, alongside an analysis of the computational resources and hardware required for effective deployment. This approach ensures that each node aligns with system requirements, supporting the deployment of PLIADES across various domains with optimal resource utilization.

A diagrammatic illustration of this methodology is provided in the following figure (Figure 2).

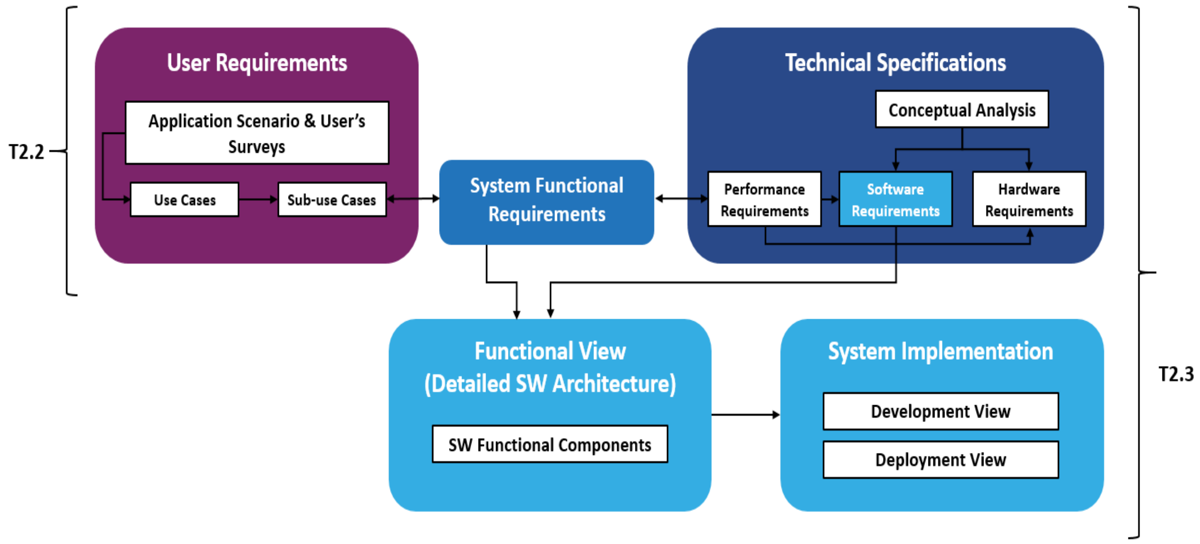

Figure 2 Diagram of the Employed Architectural Design Process

2.3 Methodology Overview for System Definition and Technical Specifications

To provide a comprehensive understanding of the purpose of deliverable D2.3 and its connection to D2.2, it is essential to outline the initial stages of the PLIADES platform's development process. The methodology encompasses the following steps:

- **User Needs:** Derived from the overarching project vision, user surveys, and a thorough analysis of user requirements, these insights lead to the identification of use cases and sub-use cases, which will be documented within deliverable D2.2.
- **Functional Design:** Based on sub-use cases, the PLIADES platform's functional design is created, resulting in a detailed set of functional requirements to directly address the defined user needs.
- **Architectural Design:** The functional requirements are combined with available technology resources (e.g., technologies available to the consortium or prior results) and design constraints (such as regulatory, safety and environmental standards). This integration forms the foundation for the architectural design of the hardware and software components.
- **Technical Specifications:** The architectural design provides a structured basis for the development of technical specifications, covering hardware, software, and performance requirements. These specifications are prioritized and systematically mapped to user requirements, ensuring alignment with project objectives and facilitating efficient tracking for implementation.

2.3.1 Mapping of User Requirements to Technical Specifications

As part of the methodology outlined in Section 2.3, the performance requirements for the PLIADES platform are meticulously derived by mapping user needs to technical specifications. This process begins with an analysis of each use case and sub-use case to ensure alignment with specific user requirements. Through this analysis, the performance requirements for the PLIADES platform are established. Technical specifications are then defined, focusing on the software components and any supporting hardware that meet both functional and performance requirements. This structured mapping process lays the groundwork for the system's technical development, aligning user requirements with system capabilities. The overall workflow for defining the technical specifications is depicted in Figure 3.

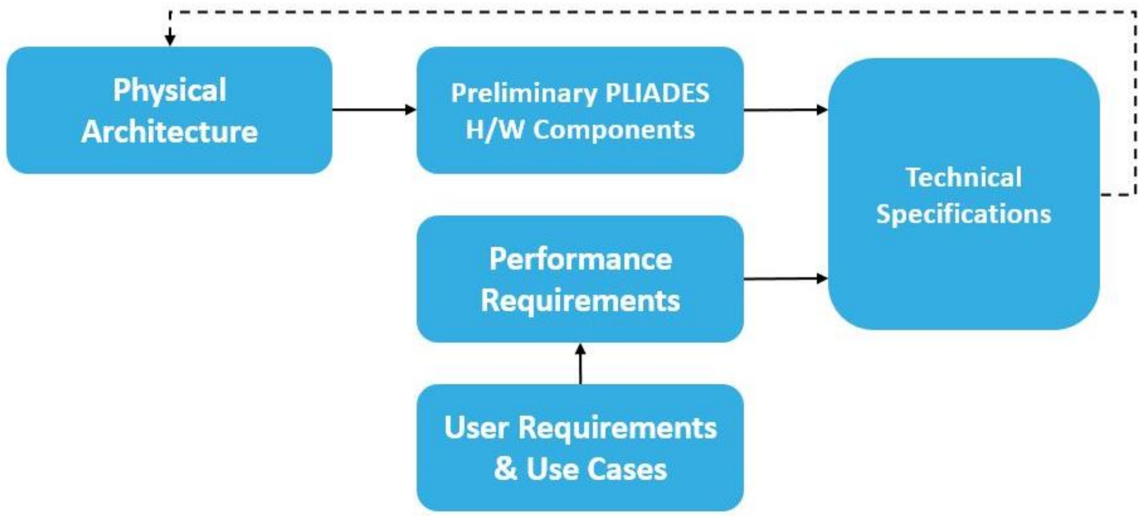

Figure 3 Workflow followed for the definition of Technical Specification of the PLIADES system

2.3.2 Definition of the PLIADES Physical Architecture

To establish the physical architecture for the PLIADES system, essential hardware components—including data acquisition sensors, communication interfaces, and processing units—are specified according to performance needs and the intended data integration setup. Subsequently, these components are mapped to ensure they integrate seamlessly within the system's conceptual design. This step identifies necessary interfaces and connectivity options to facilitate smooth data flow and communication across nodes.

2.3.3 Establishment of the Technical Specifications of the H/W components of PLIADES

With a preliminary design of the physical architecture established, project partners collaborate to finalize specifications for each hardware component in the PLIADES system. This involves selecting commercially available components that best align with system requirements, focusing on compatibility with software modules and data interoperability standards. The collaborative approach ensures each hardware element meets the project's performance standards and functional requirements, supporting seamless integration and deployment across diverse operational scenarios.

3 Conceptual Architecture Analysis

3.1 Introduction

The PLIADES project aims to establish a robust framework for data lifecycle optimization and cross-domain data integration, using AI advancements and advanced interoperability solutions to drive technological progress in AI and robotics. The project focuses on building a multi-layered architecture to foster data-driven innovation across diverse sectors such as mobility, energy, industry, healthcare and green deal.

To achieve this vision, the PLIADES platform integrates novel AI-enabled tools to advance full data lifecycle management within and between data spaces. Sustainable data creation methods—such as data compression, filtering, and normalization—are employed to support efficient and eco-friendly data storage, promoting a sustainable data-oriented future. Ensuring data privacy and sovereignty is a core priority, achieved through adherence to data standards and decentralized protocols that safeguard both data-producing organizations and individuals.

A core component of the PLIADES architecture is an advanced, AI-powered data broker, designed to dynamically manage cross-domain data sharing. This broker, along with AI-based connectors, leverages extended metadata and best practices shaped by domain expertise to revolutionize data sharing. Through active data discovery services and cross-domain AI connectors, PLIADES creates linked data spaces that enhance interoperability and enable seamless integration of previously isolated entities.

The PLIADES platform supports this multi-domain integration through a diverse set of modular tools and sensors, including IoT devices and edge-computing capabilities for real-time data capture and processing. Explainable AI, federated learning, and privacy-preserving data-sharing mechanisms ensure secure and efficient data integration, supported by a robust data governance framework that enforces strict security and compliance standards.

Data quality assessment services are also embedded within the system, providing real-time evaluations that enhance decision-making capabilities across sectors. Additionally, PLIADES establishes synergies with EU initiatives, contributing models, strategies, and technologies to the Common European Data Space.

The PLIADES outcomes are validated across six use cases focused on direct advancements in AI and robotics technologies within the targeted data spaces. These use cases present a complex validation environment that addresses full data lifecycles, encompassing vast, heterogeneous data creation, management, and sharing across multiple domains. The resulting PLIADES ecosystem aims to enhance Connected Cooperative and Automated Mobility (CCAM) and ADAS/AD technologies, reshape Human-Robot Interaction (HRI) for both operators and healthcare patients, and deploy integrated data spaces across healthcare, manufacturing, and Green Deal sectors. This approach not only contributes to reducing carbon footprints, but also lays the groundwork for a greener, data-centric future.

3.2 PLIADES Architecture Overview

The architecture is organized into four **technical layers**:

1. **Data Creation Layer:** Focuses on generating high-quality data across domains, incorporating AI-driven data elaboration, human-knowledge injection, and digital twin implementations.
2. **Data Processing and Analytics:** This layer provides tools for data aggregation, analytics, and visualization, supported by explainable AI models and federated learning. An AI-enabled data

quality assessment ensures that processed data meets high standards of reliability and relevance.

3. **Data Space Integration and Data Sharing:** Facilitates interoperability between different data spaces. AI-enhanced brokers and connectors manage secure data transfer and sharing strategies, allowing for seamless cross-domain integration.
4. **Data Management and Security:** Ensures data integrity and protection through on-board and cloud integration, privacy-preserving data-sharing mechanisms, and secure data management practices. This layer addresses regulatory requirements and supports compliance with EU data protection standards.

Additionally, three **complementary processes** support the technical layers:

- **Application Areas and Third-party Marketplaces/EU Initiatives:** Allows PLIADES to connect with external platforms and benefit from ongoing European digital initiatives.
- **Project Pilots and End-Users:** Real-world pilots validate the architecture and provide feedback from end-users to ensure practical applicability.
- **Data Spaces Interconnectivity:** Facilitates smooth data exchange across domains, enabled by the technical layers and the common data broker.

The PLIADES framework combines existing background technologies from project partners, newly developed assets tailored to project needs, and adapted AI models to provide a comprehensive suite of services. This architecture enables real-time data sharing, advanced analytics, and sustainable data management, thereby positioning PLIADES as a cornerstone for digital transformation across multiple sectors.

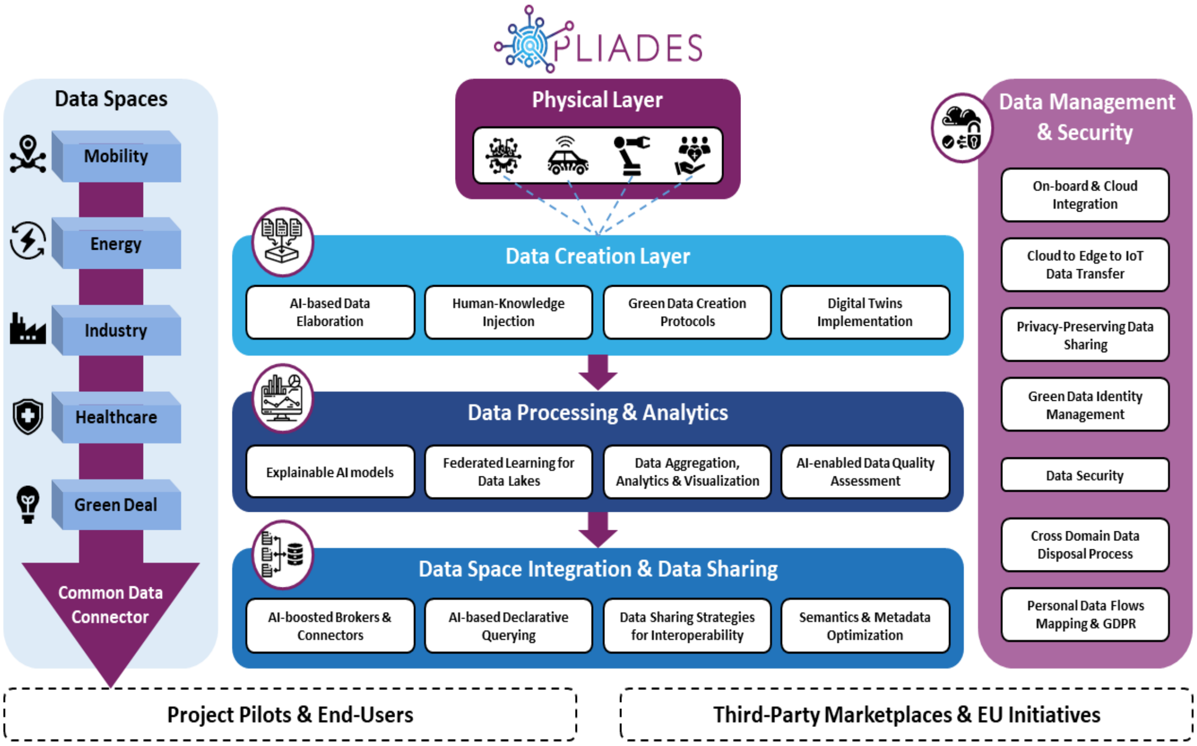

Figure 4 Overview of Conceptual Architecture of the PLIADES platform

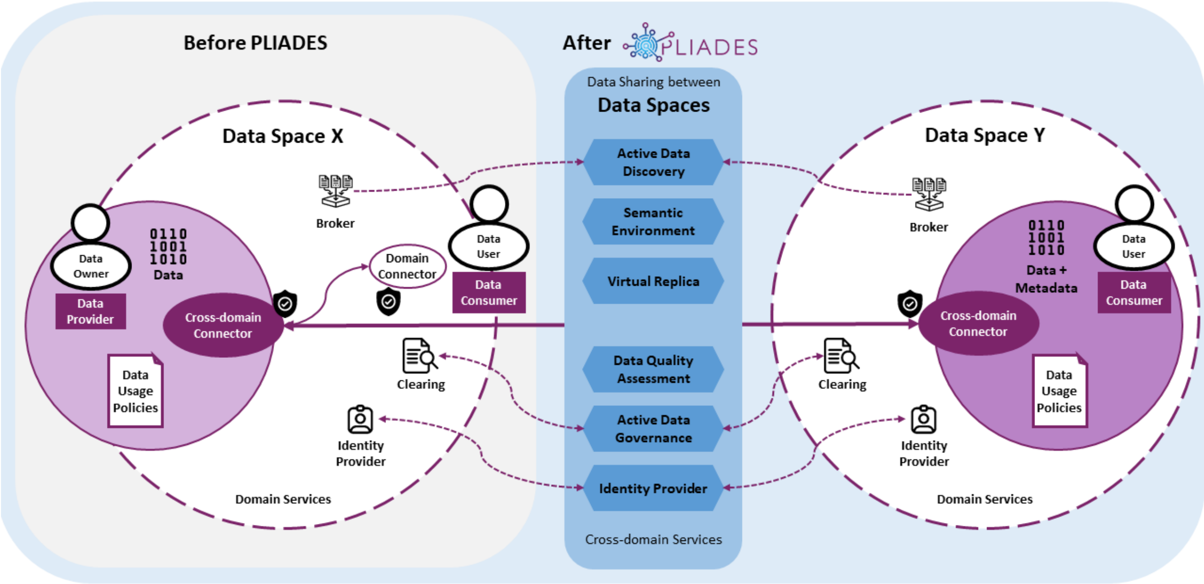

Figure 5 Conceptual Architecture of the PLIADES platform

3.3 The Basic Conceptual Modules

The PLIADES platform architecture is organized into several conceptual modules that work alongside to achieve data lifecycle optimization and data space integration. These modules address key aspects of data creation, processing, integration, and management while ensuring scalability, adaptability, and interoperability across diverse domains. Below are the primary conceptual modules:

3.3.1 Data Creation and Enhancement

The Data Creation and Enhancement module is pivotal in generating high-quality datasets across diverse domains. This module integrates AI-driven data elaboration techniques to ensure that the generated data is both accurate and comprehensive. By incorporating human knowledge, the system leverages domain expertise to refine and validate data, ensuring contextual relevance and reliability. Additionally, digital twin implementations provide virtual representations of real-world systems, enabling continuous simulation and analysis for optimizing data creation processes. These advanced mechanisms ensure that data produced within the PLIADES framework meet the requirements for high fidelity and usability, setting the foundation for efficient data processing and analytics extraction.

3.3.2 Data Processing and Analytics Extraction

The Data Processing and Analytics Extraction module delivers advanced tools and methodologies for aggregating, analyzing, and visualizing data. Main part of this module is the explainable AI models that offer interpretable results, enabling users to gain actionable insights. Federated learning techniques are employed to allow collaborative model training across distributed datasets while preserving data privacy. An AI-enabled data quality assessment ensures the processed data adheres to stringent standards of accuracy and relevance. Visualization tools further enhance the usability of this module, allowing stakeholders to explore data relationships and trends effectively. Together, these features empower users to derive meaningful analytics and informed decision-making.

3.3.3 Data Space Integration and Data Sharing

The Data Space Integration and Data Sharing module facilitates seamless interoperability between diverse data spaces. AI-enhanced brokers and connectors ensure secure and efficient data transfer, enabling smooth cross-domain data integration. Semantic interoperability tools provide a consistent framework for interpreting data across varying domains, ensuring that data retains its meaning and usability. Virtual replicas simulate data structures to test and optimize integration strategies. This module also incorporates cross-domain integration services, which streamline collaboration and information exchange, making the PLIADES framework a cornerstone for interconnected data ecosystems.

3.3.4 Data Management and Security Assurance

The Data Management and Security Assurance module ensures the integrity, privacy, and compliance of data within the PLIADES framework. It employs privacy-preserving mechanisms such as differential privacy and encryption to safeguard sensitive information during processing and sharing. On-board and cloud integration solutions provide flexibility and scalability for managing data across different environments. Compliance tools ensure adherence to EU data protection regulations, such as GDPR, while secure data governance frameworks enforce access control, integrity, and traceability. This module guarantees that data is managed securely and responsibly, meeting the highest standards of reliability and trustworthiness.

4 Use Case-Specific Functional Analysis

4.1 Use Case Analysis

As outlined in deliverable D2.2 "User Requirements and Use Cases", each use case has been divided into scenario-based sub-use cases, detailing the specific activities that the PLIADES platform is expected to perform. Additionally, user requirements have been associated with one or more sub-use cases, as illustrated in the table below (Table 2).

Table 2 Summary of the High-level Use Cases and their Sub-Use Cases

High Priority	Medium Priority	Low Priority
High-level Use Case	**Sub-Use Case ID and Title**	**Related requirements**
***UC1**. Integrating data life cycles of sustainability, operations and process industry manufacturing operations (Industrial Data Space)*	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization	***Functional*** *UR1, UR2, UR3, UR4, UR5, UR6, UR7, UR8, UR9, UR10, UR11, UR12, UR13, UR14, UR15* ***Non-functional*** *UR62, UR63, UR64, UR65, UR66, UR67, UR68, UR69, UR70, UR71, UR72, UR73, UR74, UR75, UR76*
***UC2**. Integrating data life cycles of service robot to improve HRI with end users (Healthcare Data Space)*	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring	***Functional*** *UR16, UR17, UR18, UR19, UR20, UR21, UR22, UR23, UR24, UR25, UR26, UR27, UR28, UR29* ***Non-functional*** *UR78*
***UC3**. Integrating data life cycles of personalized medicine services to improve diagnostic and prognostic clinical prediction models (Healthcare Data Space)*	SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition	***Functional*** *UR26, UR28, UR29, UR30, UR31, UR32, UR33* ***Non-functional*** *UR66, UR78, UR79*
***UC4**. Integrating data life cycles of smart vehicles for CCAM*	SubUC4.1. AI-based ADAS development	***Functional***

High Priority	Medium Priority	Low Priority
High-level Use Case	**Sub-Use Case ID and Title**	**Related requirements**
operations and ADAS/AD functions (Mobility and Industrial Data Spaces)	SubUC4.2. AI-based traffic management	*UR16, UR18, UR25, UR27, UR34, UR35, UR36, UR37, UR38, UR40, UR41, UR42, UR43, UR44, UR45, UR46, UR47, UR48, UR49, UR50, UR51, UR52, UR53* ***Non-functional*** *UR77, UR80, UR81, UR82, UR83, UR84, UR85, UR86, UR87, UR88, UR89, UR90*
***UC5**. Integrating data life cycles of WEEE/batteries management and car parts manufacturing operations (Green Deal and Industrial Data Spaces)*	SubUC5.1. Sharing field data from battery systems energy storage installations	***Functional*** *UR54, UR55, UR56, UR57, UR58, UR59, UR60, UR61*
***UC6**. Integrating professional service robot data life cycles to improve HRI with robot operators (Healthcare and Industrial Data Spaces)*	SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	***Functional*** *UR16, UR17, UR18, UR19, UR20, UR21, UR22, UR23, UR24, UR25, UR26, UR27, UR28, UR29* ***Non-functional*** *UR77*

4.2 Functional Design of the System

System functional requirements should be established with reference to the data operations that should be achieved with software solutions in order to assign technical specifications to sub-use cases. As a result, every sub-use case has been examined to generate a list of tasks that must be completed. This section provides a summary of all system functional requirements pertaining to particular sub-use cases. Translating user requirements and use cases into precise technical specifications is the primary goal of the system functional requirements. Additionally, it must be noted that the hardware and software requirements for each sub-use case, the technical specifications, are the ones that will be developed and implemented throughout the PLIADES project.

In order to identify the functionalities that will unavoidably be repeated in numerous sub-use cases, the list of identified functional requirements was initially kept as abstract as possible. As a result, each system functional requirement can be used in multiple sub-use cases, as indicated in the following table (Table 3).

Table 3 Summary of the identified system functional requirements and their respective sub-use cases.

No. /ID	System Functional Requirement	Related Sub-Use Cases	Feasibility/ Relevance
Data Collection Requirements			
SFR1	Collect raw data from multiple sources	SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH
SFR2	Collect raw data from multiple agents	SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation	LOW

		SubUC6.3. HRI in manufacturing inspection robot operation	
Data Synchronization Requirements			
SFR3	Synchronize data based on timestamps	SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH
SFR4	Synchronize data based on location	SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	LOW
Data Preprocessing Requirements			
SFR5	Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics	SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC2.1. HRI in rehabilitation	HIGH

		SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	
SFR6	Prioritize data based on their quality metrics	SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	MEDIUM
SFR7	Apply harmonization pre-processing techniques to ensure data consistency	SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC2.1. HRI in rehabilitation	HIGH

		SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations	
SFR8	Perform ad-hoc analytics pre-processing and variable engineering methods	SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC5.1. Sharing field data from battery systems energy storage installations	MEDIUM
Data Annotation Requirements			
SFR9	Generate annotation labels for data in a semi-automated manner	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations	MEDIUM

		SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	
SFR10	Extract automated labels for datasets assigned with confidence level	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC5.1. Sharing field data from battery systems energy storage installations	MEDIUM
SFR11	Allow human validation of dataset labels with low confidence	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH
SFR12	Inject human knowledge into data annotation and metadata creation processes	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation	HIGH

		SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	
SFR13	Annotate multimodal data	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH
SFR14	Visualize existing annotations of multimodal data	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	MEDIUM
Data Transformation and Conversion			
SFR15	Convert data between different formats	SubUC2.1. HRI in rehabilitation	HIGH

		SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	
SFR16	Convert datasets into actionable data products	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC5.1. Sharing field data from battery systems energy storage installations	HIGH
AI Model Improvement and Monitoring			
SFR17	Establish automated monitoring and evaluation mechanisms for tracking AI model performance improvements	SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition	HIGH

		SubUC5.1. Sharing field data from battery systems energy storage installations	
SFR18	Interface to collect human feedback for improving AI performance	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH
SFR19	Create artificial data from raw data	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	MEDIUM
SFR20	Enable federated learning of AI models from heterogeneous data sources	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners	HIGH

		SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	
SFR21	Provide explanations to AI models' inference	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	MEDIUM
SFR22	Develop online re-training strategies for AI models	SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management	HIGH

		SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	
Data Security and Governance			
SFR23	Apply cryptographic techniques during data exchange	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH
SFR24	Analyze and track data transactions	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC3.1. Sharing Data and Models among Partners	HIGH

		SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management	
SFR25	Anonymize or pseudonymize data by removing personally identifiable information	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH
SFR26	Ensure governance and rules in the control plane	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations	HIGH

SFR27	Enforce compliance of data transactions according to legal and ownership restrictions	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations	HIGH
SFR28	Store sensitive data with restrictions	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH

SFR29	Provide decentralized data connectors for secure, private, and sovereign data transfer between cloud and edge	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management	HIGH
SFR30	Provide decentralized mechanisms for identity access management	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC5.1. Sharing field data from battery systems energy storage installations	HIGH
SFR31	Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners	HIGH

		SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	
SFR32	Provide traceability and transparency for transactions in the entire network	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation	HIGH

		SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	
SFR33	Identify security vulnerabilities in data space architecture and protocols	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH
SFR34	Identify and provide alerts for suspicious activity and potential security threads	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation	HIGH

		SubUC6.3. HRI in manufacturing inspection robot operation	
SFR35	Analyze and prioritize threats within a dataspace	SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH
SFR36	Identify personal and non-personal data	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH
SFR37	Generate data flow maps with the entire lineage of each dataset	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security	MEDIUM

		SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	
SFR38	Identify potential privacy risks and threats	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	MEDIUM
Advanced Functionalities			
SFR39	Create federated Digital Twin models	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development	HIGH

		SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations	
SFR40	Enable cross-domain data discovery based on context and predefined criteria	SubUC4.1. AI-based ADAS development SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	MEDIUM
SFR41	Identify data drifts on multimodal data and provide alerts when detected	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition	HIGH
SFR42	Provide visual analytics and decision-making capabilities	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations	HIGH

SFR43	Extract similarity metrics for multimodal data	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management	MEDIUM
SFR44	Discard data based on similarity metrics	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management	LOW
SFR45	Perform automated data disposal processes based on expiration date	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC5.1. Sharing field data from battery systems energy storage installations	MEDIUM
Ontology and Data Management			

SFR46	Allow data exchange with ontology-based components	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH
SFR47	Perform cross-domain adaptation between datasets	SubUC4.1. AI-based ADAS development SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	MEDIUM
SFR48	Store the ontology and data into a structured database	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC5.1. Sharing field data from battery systems energy storage installations	HIGH

		SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	
SFR49	Allow queries to explore existing databases and internal relations	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management	MEDIUM
SFR50	Provide strategies for data interoperability between data spaces	SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	HIGH
SFR51	Receive natural language inputs to perform queries in Graphical Databases	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management	MEDIUM

SFR52	Provide optimal recommendations between datasets and data consumer needs	SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management	MEDIUM
Continuous Improvement and Real-Time Analytics			
SFR53	Evaluate continuously the quality of data	SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management	HIGH
SFR54	Implement strategies to improve data quality	SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development	HIGH

		SubUC4.2. AI-based traffic management	
SFR55	Extract decentralized analytics in real-time	SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC5.1. Sharing field data from battery systems energy storage installations	HIGH
SFR56	Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing	SubUC1.2. Data-Driven Lifecycle Management for Suppliers SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization SubUC2.1. HRI in rehabilitation SubUC2.2. HRI in patient monitoring SubUC3.1. Sharing Data and Models among Partners SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition SubUC4.1. AI-based ADAS development SubUC4.2. AI-based traffic management SubUC5.1. Sharing field data from battery systems energy storage installations SubUC6.1. HRI in telepresence robot operation	HIGH

		SubUC6.2. HRI in rehabilitation robot operation SubUC6.3. HRI in manufacturing inspection robot operation	

5 Functional View of the PLIADES Platform

This chapter outlines the functional view of the PLIADES platform architecture, describing its components and their corresponding functionalities at the Work Package (WP) level. The functional architecture defines how the required operations and processes are implemented through modular components, focusing on their interactions and dependencies.

The functional nodes are categorized by WP level as follows:

- **Functional Nodes for WP3-level**
- **Functional Nodes for WP4-level**
- **Functional Nodes for WP5-level**
- **Functional Nodes for WP6-level**
- **Functional Nodes that Interact Directly with the End-User**

In the accompanying diagrams, the tasks related to each WP are color-coded for clarity: WP3 tasks are represented in **green**, WP4 tasks in **blue**, WP5 tasks in **red**, and WP6 tasks in **purple**.

The PLIADES platform consists of the 27 functional nodes, which are listed below:

- **Functional Nodes for WP3-level**
    - Green Data Generation (T3.1, T3.2)
    - Data harmonization and Pipelines Orchestration (T3.3)
    - Active Learning (T3.4)
    - Data Elaboration (T3.5)
    - Generative AI (T3.6)
    - Compliant Data Governance with Quality (CompliantDQ) (T3.7)
- **Functional Nodes for WP4-level**
    - Hyperledger Fabric Blockchain – Data Space Connector Interface (T4.1)
    - Cloud to Edge to IoT transfer (T4.2)
    - Privacy-Enhanced Data Collaboration (PEDC) (T4.3)
    - Identity Access Management- Data Space Connector Decentralization (T4.4)
    - Federated Data Security (FDS) (T4.5)
    - Relevant Data Selection (T4.6)
    - Personal Data Guard (T4.7)
- **Functional Nodes for WP5-level**
    - Proven WebLabel Solution (T5.1)
    - Proven Ontology Manager (T5.1)
    - Interoperability Sharing Strategies/ Protocol (T5.2)
    - AI-based Query System (T5.3)
    - Active Data Discovery (T5.4)
    - Human and Data Centric AI Model Performance Evaluation (T5.5)
    - AI Connector (T5.6)
- **Functional Nodes for WP6-level**
    - Explainable Model Creation (T6.1, T6.2)
    - Online Retraining (T6.1, T6.2)
    - Federated learning (FL) (T6.3)
    - Data Aggregation, Analytics, and Visualization (DAAV) (T6.4)
    - Data Re-use System for Decision Support Systems (T6.5)
    - AI-Enabled Data Quality Management (AI-DQM) (T6.6)
    - Federated cross-domain data Analytics (T6.7)
- **Functional Nodes that Interact Directly with the End-User**

5.1 Functional Nodes from WP3 vantage point

The functional nodes from WP3 vantage point are presented in the following diagram (Figure 6).

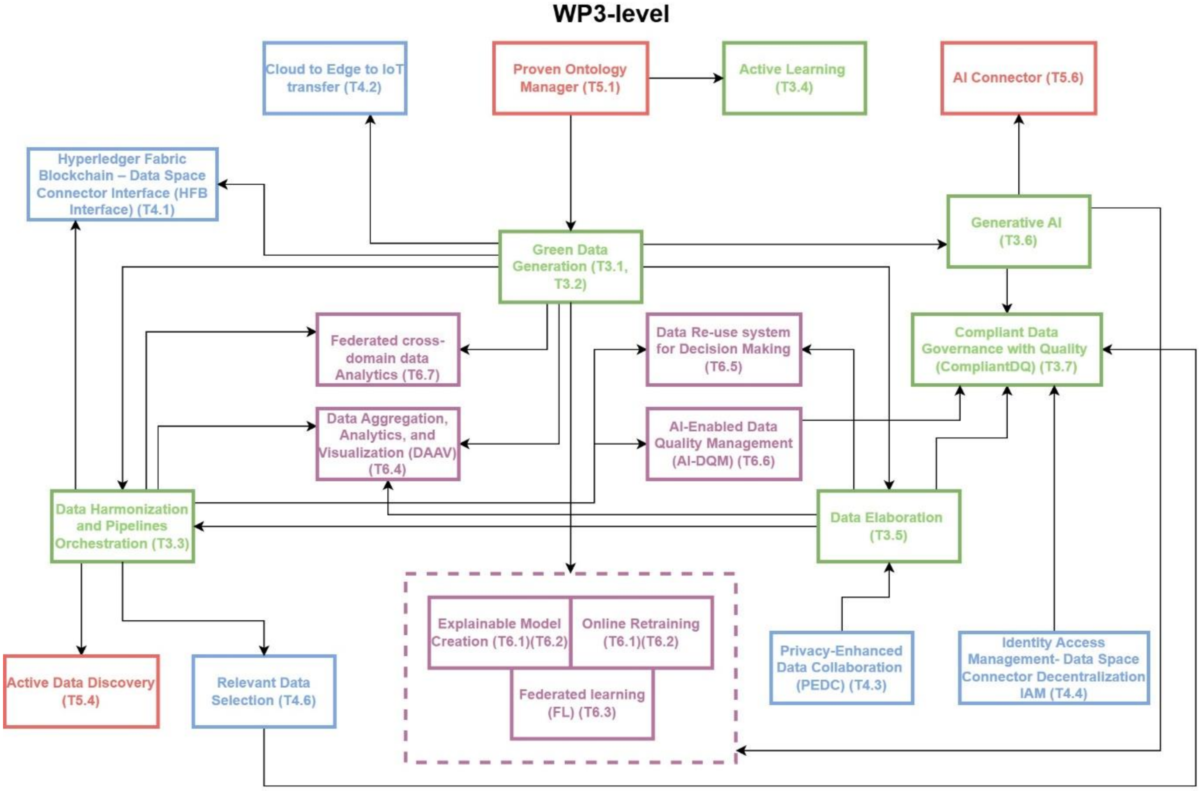

Figure 6 WP3-level Architecture

5.1.1 Green Data Generation (T3.1, T3.2)

5.1.1.1 Description

The Green Data Generation node is responsible for collecting raw and multimodal sensor data from multiple sources and agents (robots, vehicles, fixed infrastructure, wearables, HRI), synchronizing them based on timestamps and location data, and finally filtering out noise, outliers and discrepancies, towards greener data practices. Also, this node will cover the labelling of specific data, creating complete datasets, which includes both high quality raw data and annotations for specific tasks. The overall aim of this node is to convert all data in a common format suitable for efficient data sharing and interoperability among data spaces.

5.1.1.2 Main Functionalities

- Generation of a synchronized dataset in terms of date, time and location.
- Grouping and categorization using the corresponding metadata.
- Prioritization of data based on quality metrics.

5.1.1.3 Internal Components

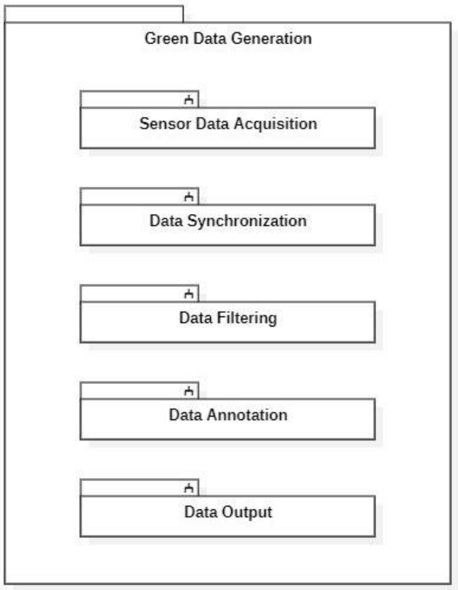

Figure 7 Green Data Generation node – Internal Components

The Green Data Generation module consists of the following internal components:

- ***Sensor Data Acquisition***

  Sensor Data Acquisition node subscribes to various data sources like onboard LiDAR and camera (RGB and/or depth), fixed sensors, wearables, etc. Each sensor might have its own communication protocol or API for data retrieval, such as ROS for robotic agents.

- ***Data Synchronization***

  Data Synchronization module receives as input multimodal data, such as images, LiDAR measurements, real-time analytics, etc., and synchronizes them according to the relevant metadata, including location, timestamp and other environmental conditions. Algorithms like time interpolation or correlation might be used to compensate for minor timing discrepancies between sensors.

- ***Data Filtering***

  Data Filtering node receives the synchronized data as input and performs basic filtering steps in order to eliminate low quality data, remove redundant information and apply basic curation steps. Filtering techniques might include statistical analysis, outlier detection algorithms, and spatial consistency checks.

- ***Data Annotation***

  Data Annotation node receives the filtered data and performs semi-automated dataspace-specific annotation, generating labels related to the dataspace domain, such as semantic annotations on images, basic metadata addition, etc.

- ***Data Output***

  Data Output node provides the processed, filtered and annotated data to other external systems/nodes for further processing or analysis. The output format might be a unified data structure or individual data streams depending on the agreed dataset format.

5.1.1.4 Dependencies to other Components

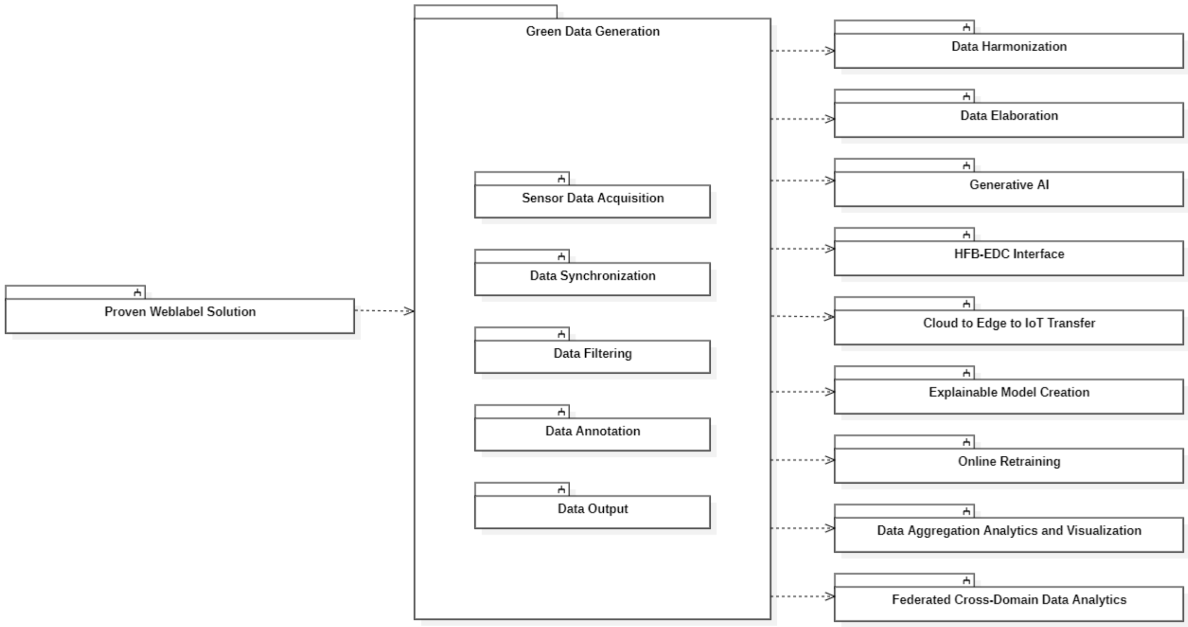

Figure 8 Green Data Generation node – Connection with other components

Green Data Generation node interacts with the following components:

- ***Proven Weblabel Solution (T5.1) (From)***

The Green Data Generation node uses the Proven Weblabel Solution in order to generate annotations for the generated datasets, according to the use-case and the dataspace needs.

- ***Data Harmonization (T3.3) (To)***

The Green Data Generation node provides input to the Data Harmonization node, which will be responsible to perform further harmonization steps and generate data products based on the created datasets.

- ***Data Elaboration (T3.5) (To)***

The Green Data Generation node provides input to the Data Elaboration node, which will use the generated datasets from different dataspaces and scenarios to develop cross-domain adaptation methods and data virtualization techniques for multimodal sources.

- ***Generative AI (T3.6) (To)***

The Green Data Generation node provides input to the Generative AI node, which will use raw data to create artificial ones.

- ***HFB-EDC Interface (T4.1) (To)***

The Green Data Generation node provides input to the HFB-EDC Interface the created data models to be integrated with the EDC data plane.

- ***Cloud to Edge to IoT Transfer (T4.2) (To)***

The Green Data Generation node provides as input to the Cloud to Edge to IoT transfer node the created metadata, operational data and user data from IoT sensors, edge devices and cloud platforms.

- ***Explainable Model Creation (T6.1, T6.2) (To)***

The Green Data Generation node provides raw data to the Explainable Model Creation node, in order to develop Explainable AI models.

- ***Online Retraining (T6.1, T6.2) (To)***

The Green Data Generation node provides datasets to the Online Retraining node, in order to develop and perform continuous retraining of AI models using updated data.

- ***Data Aggregation, Analytics, and Visualization (DAAV) (T6.4) (To)***

The Green Data Generation node provides data to the Data Aggregation, Analytics, and Visualization (DAAV) node for the generation of data analytics and insightful visualizations.

- ***Federated Cross-Domain Data Analytics (T6.7) (To)***

The Green Data Generation node provides processed real data and raw data from the real world to the Federated Cross-Domain Data Analytics node for the extraction of data analytics among different domains.

5.1.1.5 Addressed System Functional Requirements

SFR ID	Description
SFR1	Collect raw data from multiple sources
SFR2	Collect raw data from multiple agents
SFR3	Synchronize data based on timestamps
SFR4	Synchronize data based on location
SFR5	Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics
SFR6	Prioritize data based on their quality metrics
SFR10	Extract automated labels for datasets assigned with confidence level
SFR13	Annotate multimodal data
SFR15	Convert data between different formats
SFR25	Anonymize or pseudonymize data by removing personally identifiable information
SFR28	Store sensitive data with restrictions
SFR44	Discard data based on similarity metrics
SFR45	Perform automated data disposal processes based on expiration date

5.1.2 Data Harmonization and Pipelines Orchestration (T3.3)

5.1.2.1 Description

The Data Harmonization and Pipelines Orchestration node is in charge of converting the data generated in the PLIADES architecture (Green Data Generation node and Data Elaboration node) and harmonizing and orchestrating it, including the expert knowledge necessary in an automated manner. The final objective is to generate data products that can be consumed and exploited by other layers of the architecture.

5.1.2.2 Main Functionalities

- Data pipelines orchestration, including DataOps governance in three points: quality, harmonization – pre-processing, integration - fusion.
- Enable ad-hoc analytics pre-processing, variable engineering definition.
- Data product standard definition and GAIA-X compliance.

5.1.2.3 Internal Components

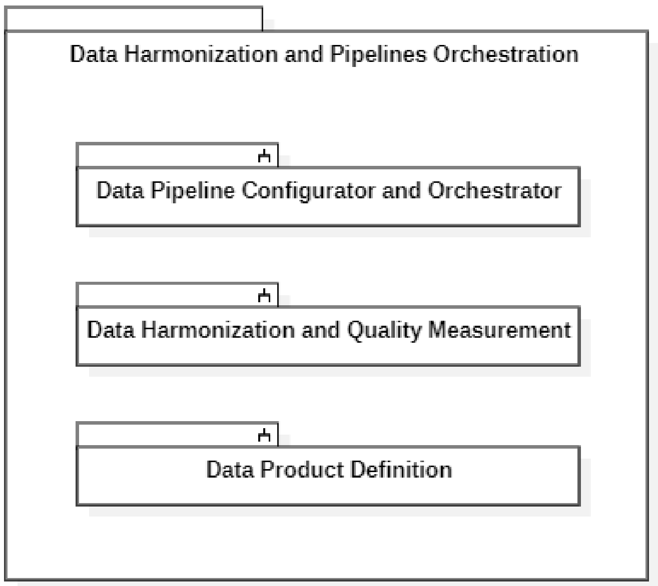

Figure 9 Data Harmonization and Pipelines Orchestration – Internal Components

The Data Harmonization and Pipelines Orchestration node consists of the following internal components:

- ***Data Pipeline Configurator and Orchestrator***

  This component automates the harmonization process in the form of DataOps pipelines, configuring them in a way that includes the necessary expert knowledge and good data governance practices.

- ***Data Harmonization and Quality Measurement***

  This component performs ad-hoc analytics pre-processing and variable engineering methods, such as the creation of new information, temporal alignment of variables, automatic determination of the importance of variables, different information homogenization techniques, automatic imputation of variables determined as anomalous/absent, generalization of variables, etc. In addition, it will measure basic quality metrics related to the applied methods that help in subsequent analysis nodes.

- ***Data Product Definition***

This component oversees exposing the datasets generated into valid data products and ensures the integration into federated data spaces.

5.1.2.4 Dependencies to other Components

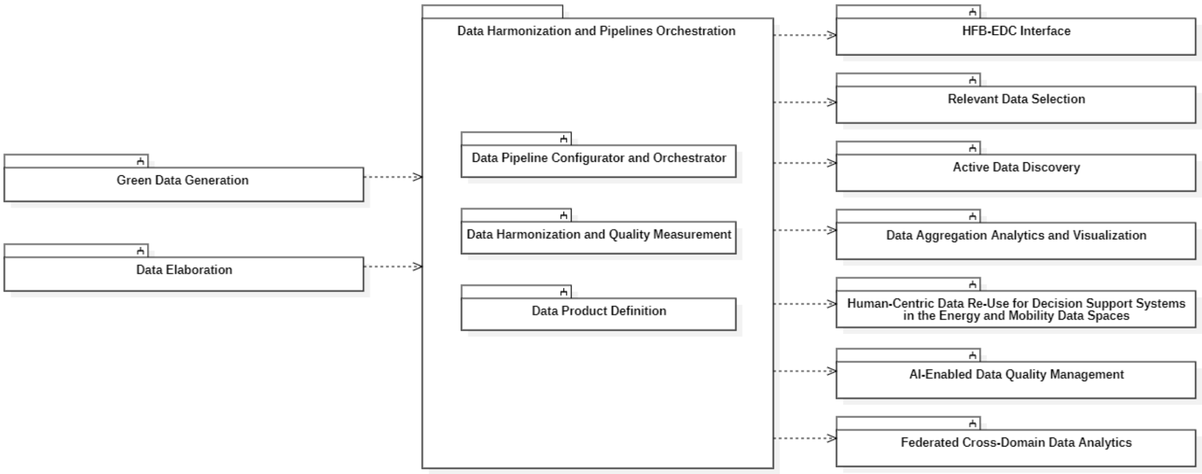

Figure 10 Data Harmonization and Pipelines Orchestration – Connection with other Components

Data Harmonization and Pipelines Orchestration node interacts with the following node:

- ***Green Data Generation (T3.1, T3.2) (From)***

The Data Harmonization and Pipelines Orchestration node receives the generated data from Green Data Generation node and prepares it to become the datasets and data products expected by the processing and analysis layers.

- ***Data Elaboration (T3.5) (From)***

The Data Harmonization and Pipelines Orchestration node receives the generated data by the Data Elaboration node and prepares it to become the datasets and data products expected by the processing and analysis layer, having cross-domain metadata.

- ***HFB-EDC Interface (T4.1) (To)***

The Data Harmonization and Pipelines Orchestration provides data products to the HFB-EDC Interface to be integrated with the EDC data plane.

- ***Relevant Data Selection (T4.6) (To)***

The Data Harmonization and Pipelines Orchestration provides quality metrics to the Relevant Data Selection node.

- ***Active Data Discovery (T5.4) (To)***

The Data Harmonization and Pipelines Orchestration provides quality metrics to the Active Data Discovery node that will be used by the AI broker to provide better suggestions.

- ***Data Aggregation, Analytics, and Visualization (DAAV) (T6.4) (To)***

The Data Harmonization and Pipelines Orchestration node provides data products to be used by the Data Aggregation, Analytics, and Visualization node.

- ***5.4.5 Data Re-use System for Decision Support Systems (T6.5) (To)***

The Data Aggregation, Analytics, and Visualization node provides data products to the Data Handling and Manipulation internal component of the node 'Data Re-use System for Decision Support Systems'.

- ***AI-Enabled Data Quality Management (AI-DQM) (T6.6) (To)***

The Data Harmonization and Pipelines Orchestration node will provide data to the AI-DQM node, so that they will be evaluated for their quality.

- ***Federated Cross-Domain Data Analytics (T6.7) (To)***

The Data Harmonization and Pipelines Orchestration node will provide data products to the Federated cross-domain data analytics node to extract analytics on these data.

5.1.2.5 Addressed System Functional Requirements

SFR ID	Description
SFR1	Collect raw data from multiple sources
SFR3	Synchronize data based on timestamps
SFR5	Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics
SFR6	Prioritize data based on their quality metrics
SFR7	Apply harmonization pre-processing techniques to ensure data consistency
SFR8	Perform ad-hoc analytics pre-processing and variable engineering methods
SFR12	Inject human knowledge into data annotation and metadata creation processes
SFR16	Convert datasets into actionable data products

5.1.3 Active Learning (T3.4)

5.1.3.1 Description

The Active Learning node will provide a mechanism to inject human knowledge into the process of data annotation and metadata creation. The main objective of this node is to extract the more relevant data which requires human intervention. In this way, the human input will help the creation of better datasets that will be used for training AI models.

5.1.3.2 Main Functionalities

The node will leverage active learning approaches to enhance the process of data creation by applying a mixed automatic metadata creation with human intervention. The result is a cyclic pipeline system that will help the faster creation of datasets. This node operates on any input data coming from the sensors. Its processes include: Automatic metadata generation, metadata filtering, human intervention, AI model improvement.

5.1.3.3 Internal Components

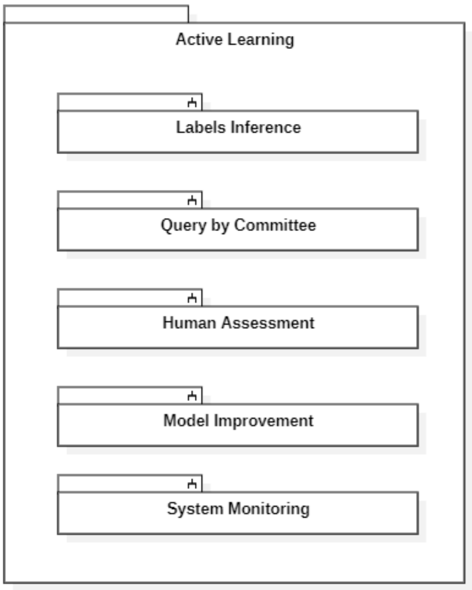

Figure 11 Active Learning – Internal Components

The active learning node will consist of the following components.

- ***Labels Inference***

This component will use an initial base AI model to perform inference on a subset of unlabelled data coming from the sensors. The output of this inference process will be formatted in a common format that will be understood by the rest of the components.

- ***Query by Committee (QbC)***

This component will rely on the assessment of different SoA models functioning as a collective group of experts to make predictions. A scoring mechanism will then assign a confidence level to each predicted label. Labels with the lowest confidence scores will be flagged for human intervention.

- ***Human Assessment***

This component will allow the human to validate those labels with the lowest confidence provided by the QbC component.

- ***Model Improvement***

This component will take new labels generated by either human intervention or automatically by the QbC scoring system. Then a new AI model is trained to improve the base model used for label generation.

- ***System Monitoring***

This component will test that the new models created by the Active Learning system is not being degraded by the cyclic process. The process will check upon a previously labelled dataset to check for inconsistencies.

5.1.3.4 Dependencies to other Components

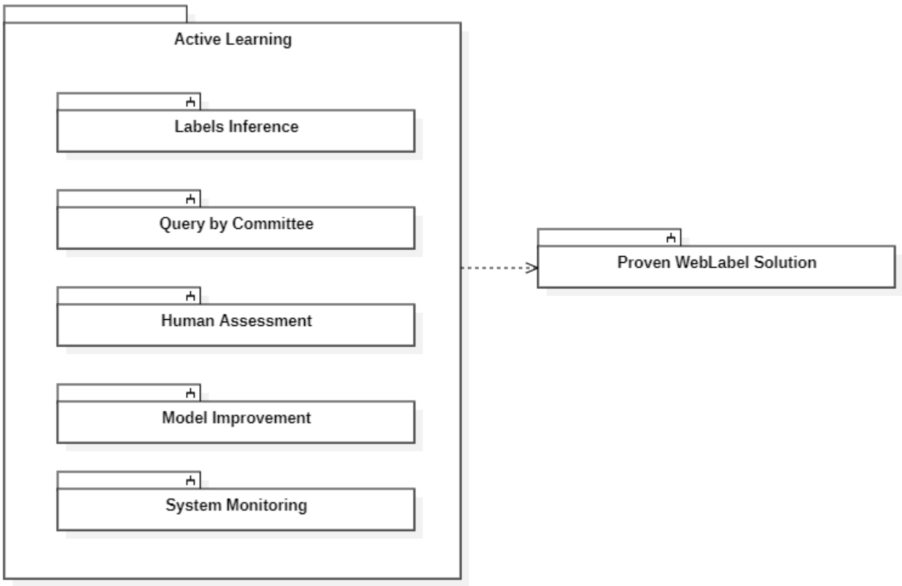

Figure 12 Active Learning – Connection with other Components

- ***Proven WebLabel Solution (T5.1) (To)***

The Proven WebLabel Solution will use the techniques developed by the Active Learning node to enhance the semi-automated annotation pipelines, as part of the annotation tool.

5.1.3.5 Addressed System Functional Requirements

SFR ID	Description
SFR9	Generate annotation labels for data in a semi-automated manner
SFR10	Extract automated labels for datasets assigned with confidence level
SFR11	Allow human validation of dataset labels with low confidence
SFR12	Inject human knowledge into data annotation and metadata creation processes
SFR13	Annotate multimodal data
SFR14	Visualize existing annotations of multimodal data
SFR18	Interface to collect human feedback for improving AI performance

5.1.4 Data Elaboration (T3.5)

5.1.4.1 Description

The Data Elaboration node is responsible for extending the capabilities of the generated data, allowing their reusability in multiple scenarios and situations. It will provide models that will allow cross-domain adaptation of data, to be used among the same or different domains, while also provide a layer to access these data from various domains.

5.1.4.2 Main Functionalities

Develop interfaces that allow data exchange with ontology-based components, based on metadata, leading to virtualization capabilities. Development of cross-domain adaptation methods.

5.1.4.3 Internal Components

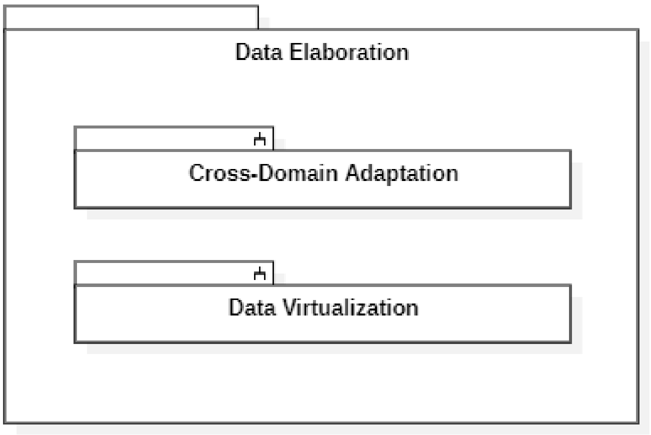

Figure 13 Data Elaboration – Internal Components

The Data Elaboration node consists of the following internal component:

- ***Cross-Domain Adaptation***

  Cross-Domain Adaptation component receives as input multimodal data, such as images, LiDAR, real-time analytics, etc., and performs AI-based techniques that allow the usage of these data in different settings (conditions, situations). For example, data of low-resolution LiDAR sensors are processed and provide data that allow inference with high-resolution LiDAR sensors. Also, data captured from the mobility domain can be adapted to be used in industrial settings, etc.

- ***Data Virtualization***

  Data Virtualization component receives as input multimodal data from heterogeneous data sources and performs all the necessary operations to create a single data-access layer for finding and accessing the data.

5.1.4.4 Dependencies to other Components

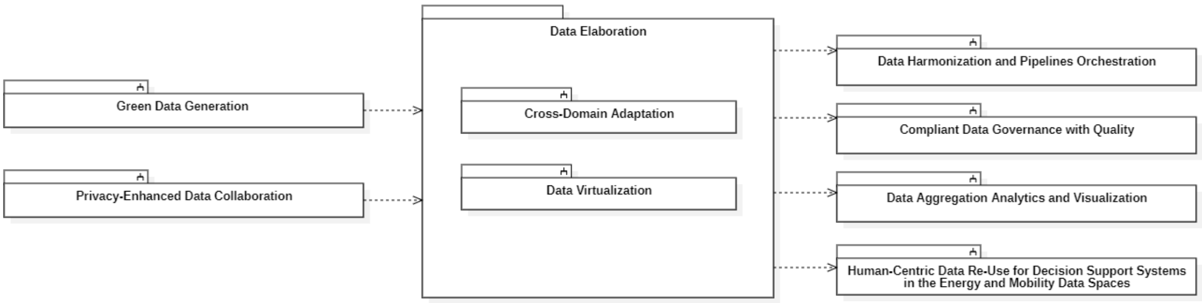

Figure 14 Data Elaboration – Connection with other Components

Data Elaboration node interacts with the following components:

- ***Green Data Generation (T3.1, T3.2) (From)***

The Data Elaboration node receives data from the Green Data Generation node , as the initial datasets with the essential labels that will be enhanced with re-use capabilities.

- ***Privacy-Enhanced Data Collaboration (PEDC) (T4.3) (From)***

The Data Elaboration node receives anonymized data from the Privacy-Enhanced Data Collaboration node (PEDC) to ensure that privacy-preserving techniques are ensured during data re-use.

- ***Data Harmonization and Pipelines Orchestration (T3.3) (To)***

The Data Elaboration node provides data to the Data Harmonization and Pipelines Orchestration node to convert the generated data to data products, so that they can be used among the data spaces.

- ***Compliant Data Governance with Quality (CompliantDQ) (T3.7) (To)***

The Data Elaboration node provides data enhanced with metadata to provide re-use capabilities, as part of the automated data governance procedures, following the quality and legislation needs.

- ***Data Aggregation, Analytics, and Visualization (DAAV) (T6.4) (To)***

The Data Elaboration node provides context-aware data and data elaboration methods to the Data Aggregation, Analytics, and Visualization (DAAV) node that will be used to generate meaningful analytics and visualizations.

- ***Data Re-use System for Decision Support Systems (T6.5) (To)***

The Data Elaboration node provides methods for data re-use capabilities to the • Data Re-use System for Decision Support Systems node, so that will assist the development of extended connectors for data reuse in energy and mobility.

5.1.4.5 Addressed System Functional Requirements

SFR ID	Description
SFR40	Enable cross-domain data discovery based on context and predefined criteria
SFR46	Allow data exchange with ontology-based components
SFR47	Perform cross-domain adaptation between datasets
SFR48	Store the ontology and data into a structured database
SFR52	Provide optimal recommendations between datasets and data consumer needs

5.1.5 Generative AI (T3.6)

5.1.5.1 Description

The Generative AI node is responsible for the development of Generative AI methods to generate artificial data from raw ones, aiming to the creation of federated Digital Twin models that will represent different domains.

5.1.5.2 Main Functionalities

Develop Generative AI methods to create artificial data. Create federated digital twin models.

5.1.5.3 Internal Components

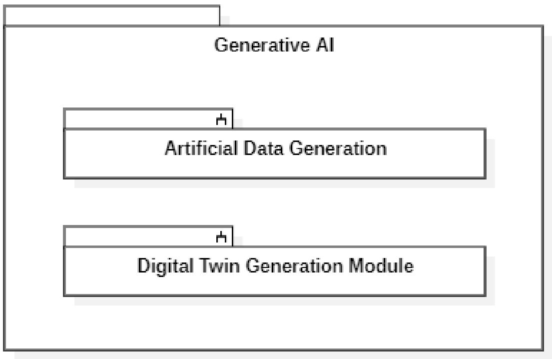

Figure 15 Generative AI – Internal Components

The Generative AI node consists of the following internal component:

- ***Artificial Data Generation***

  Artificial Data Generation component uses as input multimodal data, such as images, LiDAR, real-time analytics, text prompts, etc., in order to train generative AI methods that will generate artificial data.

- ***Digital Twin Generation Node***

  Digital Twin Generation component receives as input multimodal data, such as images, LiDAR, real-time analytics, etc., either real or artificial from simulator tools, and through federated learning techniques, it creates Federated Digital Twin models.

5.1.5.4 Dependencies to other Components

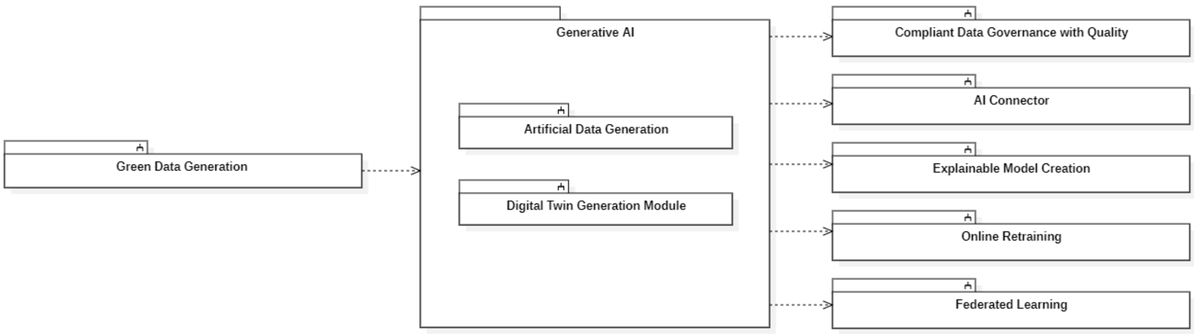

Figure 16 Generative AI – Connection with other Components

Generative AI node interacts with the following components:

- ***Green Data Generation (T3.1, T3.2) (From)***

  The Generative AI node receives data from the Green Data Generation node, required to train generative AI methods.

- ***Compliant Data Governance with Quality (CompliantDQ) (T3.7) (To)***

The Generative AI node provides data to the CompliantDQ node to be used in the development of quality and legislation compliant data governance methods.

- ***AI Connector (T5.6) (To)***

The Generative AI node provides data to the AI Connector node to enable connectors elaborate on raw data.

- ***Explainable Model Creation (T6.1, T6.2) (To)***

The Generative AI node provides vast amounts of data to the Explainable Model Creation node, which are required to train and validate the Explainable AI models.

- ***Online Retraining (T6.1, T6.2) (To)***

The Generative AI node provides data from various scenarios and conditions to the Online Retraining node in order to develop strategies for the online retraining of AI models.

- ***Federated Learning (FL) (T6.3) (To)***

The Generative AI node provides data to the Federated Learning node, since this node would benefit from accessing as much training data as possible, to help the development of the module.

5.1.5.5 Addressed System Functional Requirements

SFR ID	Description
SFR19	Create artificial data from raw data
SFR20	Enable federated learning of AI models from heterogeneous data sources
SFR39	Create federated Digital Twin models

5.1.6 Compliant Data Governance with Quality (CompliantDQ) (T3.7)

5.1.6.1 Description

This node automates data governance across separate data spaces, ensuring compliance with quality standards and legislation. It operates on the following data types:

- All data transactions within the cross-domain ecosystem
- Data ownership and access control information
- Data quality metrics and legal requirements

5.1.6.2 Main Functionalities

Analyze data transactions to set security, ownership, and architecture rules based on quality, legislation, and environmental impact. Track data movement, enforce compliance, and optimize storage/processing by filtering low-quality data.

5.1.6.3 Internal Components

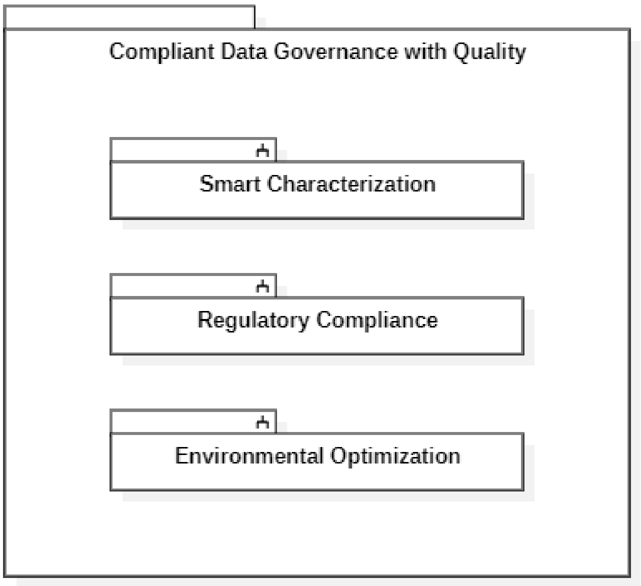

Figure 17 Compliant Data Governance with Quality – Internal Components

The node consists of the following internal components:

- ***Smart Characterization***

  Analyzes data transactions and assigns appropriate governance rules.

- ***Regulatory Compliance***

  Ensures data handling adheres to legal and ownership restrictions.

- ***Environmental Optimization***

  Minimized data footprint based on quality and retention policies.

5.1.6.4 Dependencies to other Components

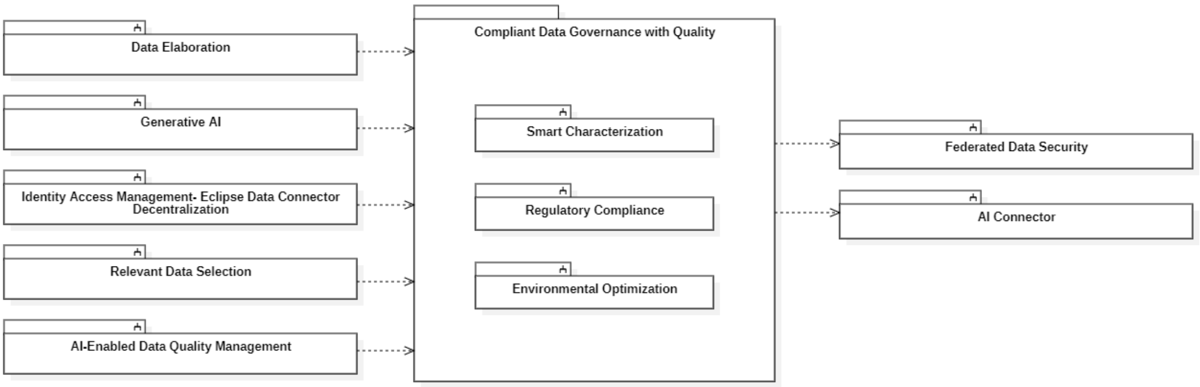

Figure 18 Compliant Data Governance with Quality – Connection with other Components

This node interacts with the following nodes:

- ***Data Elaboration (T3.5) (From)***

  The CompliantDQ node receives as input data products from the Data Elaboration node, so that these data can be used during the automated data governance procedures.

- ***Generative AI (T3.6) (From)***

The CompliantDQ AI node receives artificially generated data from the Generative AI node, so that it can operate on them for the generation of the data governance methods.

- ***Identity Access Management (IAM)- Eclipse Data Connector Decentralization (T4.4) (From)***

The CompliantDQ AI node receives the policy engine and the policies from the IAM node, to develop the regulatory compliance techniques.

- ***Relevant Data Selection (T4.6) (From)***

The CompliantDQ AI node receives quality information from the Relevant Data Selection node.

- ***AI-Enabled Data Quality Management (AI-DQM) (T6.6) (From)***

The CompliantDQ AI node receives quality assessment outcomes that will be used by the data governance methods.

- ***Federated Data Security (T4.5) (To)***

The CompliantDQ AI node provides data quality information to the Federated Data Security node to enhance thread detection.

- ***AI Connector (T5.6) (To)***

The CompliantDQ AI node provides the data governance practices as input to the AI Connector node.

5.1.6.5 Addressed System Functional Requirements

SFR ID	Description
SFR5	Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics
SFR6	Prioritize data based on their quality metrics
SFR23	Apply cryptographic techniques during data exchange
SFR24	Analyze and track data transactions
SFR25	Anonymize or pseudonymize data by removing personally identifiable information
SFR26	Ensure governance and rules in the control plane
SFR27	Enforce compliance of data transactions according to legal and ownership restrictions
SFR30	Provide decentralized mechanisms for identity access management
SFR31	Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors
SFR32	Provide traceability and transparency for transactions in the entire network

SFR36	Identify personal and non-personal data
SFR38	Identify potential privacy risks and threats
SFR44	Discard data based on similarity metrics
SFR45	Perform automated data disposal processes based on expiration date

5.2 Functional Nodes from WP4 vantage point

The functional nodes for WP4-level are presented in the following diagram (Figure 19).

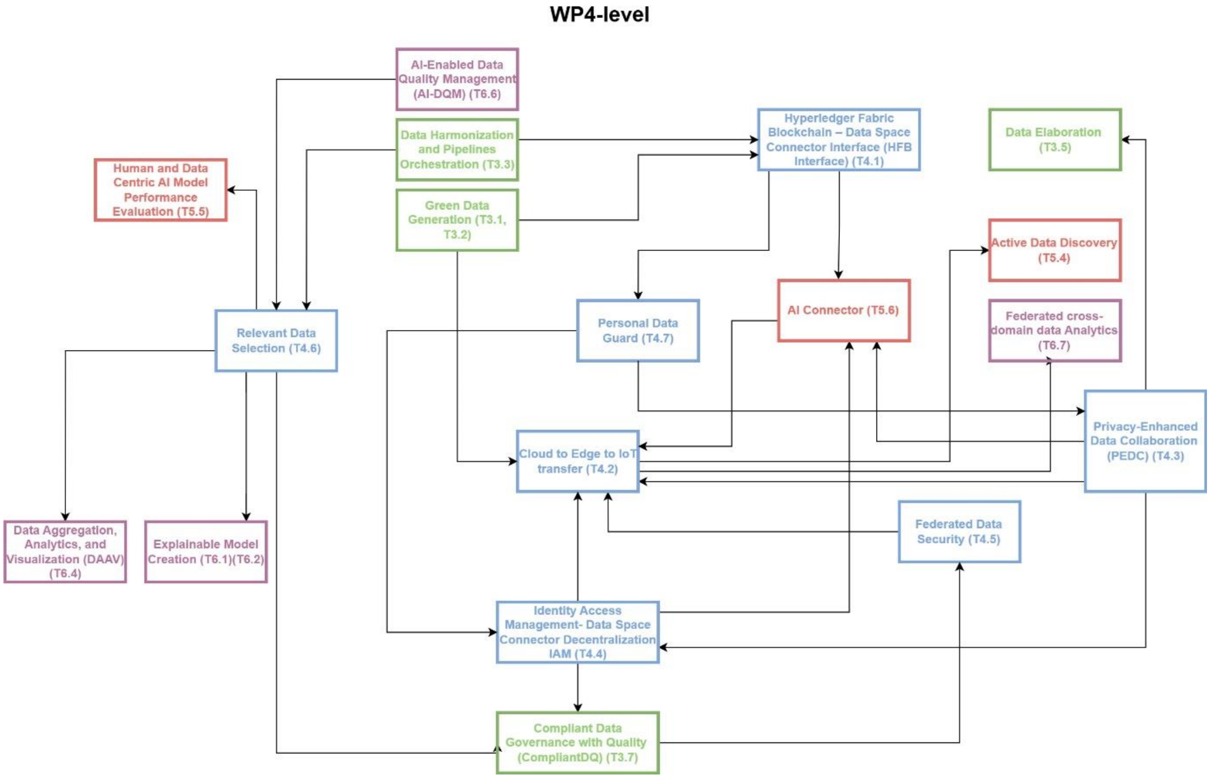

Figure 19 WP4-level Architecture

5.2.1 Hyperledger Fabric Blockchain – Data Space Connector Interface (T4.1)

5.2.1.1 Description

This node integrates blockchain technology into data spaces to ensure data integrity, establish connections between environments and facilitate their monitoring and auditing requirements. This proposes an interface-based plugin that can be integrated into an IDS Connector and allows it to communicate with Hyperledger Fabric Blockchain (HFB). From the control-plane perspective, rules and governance for such interactions can be enabled at the blockchain level, whereas from the data-plane perspective, monitoring, auditing and verifying data integrity can be enabled by the transaction-based storage mechanism of HFB. This can be complemented by Private Data Collections (PDCs) to support storing sensitive data. In particular, this software node consists of smart contracts to execute the functionalities in blockchain, client applications to expose these smart contracts to data spaces and interface-based Connector plugins to interact with these client applications.

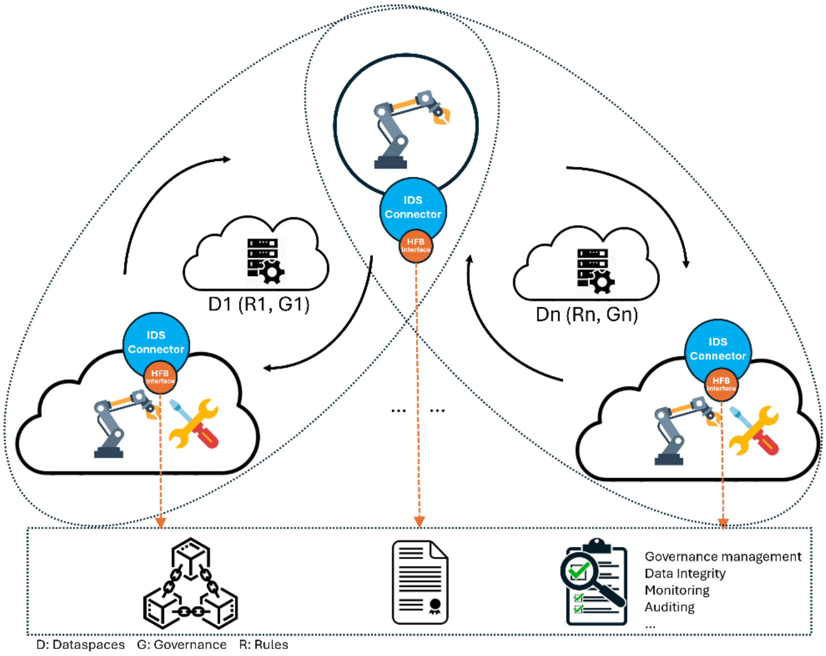

Figure 20 Integration of blockchain into data spaces through an interface

5.2.1.2 Main Functionalities

HFB Interface to ensure governance in multi-dataspace scenarios and ensure monitoring and auditing capabilities of the transmitted information. Data sensitive information will also be stored in decentralized environments while addressing and enhancing the privacy of data.

5.2.1.3 Internal Components

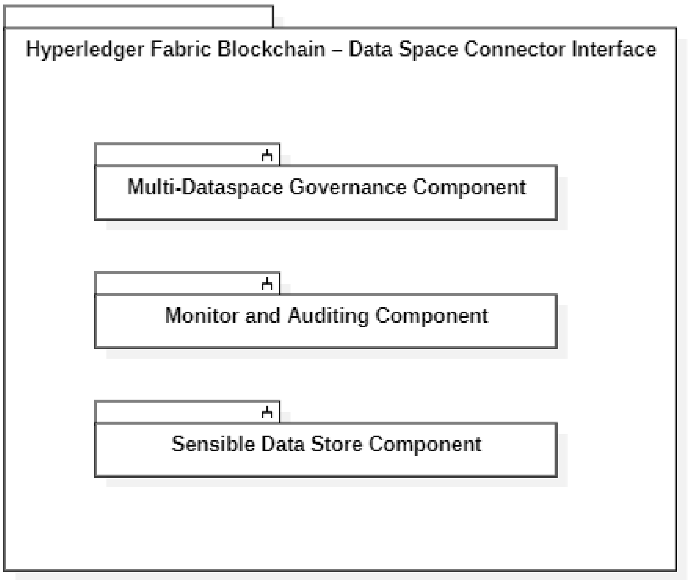

Figure 21 Hyperledger Fabric Blockchain – Data Space Connector Interface – Internal Components

The Hyperledger Fabric Blockchain Interface node consists of the following internal components:

- ***Multi-Dataspace Governance Component***

  The multi-dataspace governance component allows to verify governance and rules at the control plane level allowing the data plane to start transacting depending on the validation.

- ***Monitor and Auditing Component***

The Monitor and auditing component will enable the monitoring and auditing of an IDS Connector's transactions based on an HFB Interface and its corresponding smart contracts.

- ***Sensible Data Store Component***

The sensitive data storage component will work on the data plane enabling the ability to store sensitive data from use cases with restrictions, e.g., healthcare environment.

5.2.1.4 Dependencies to other Components

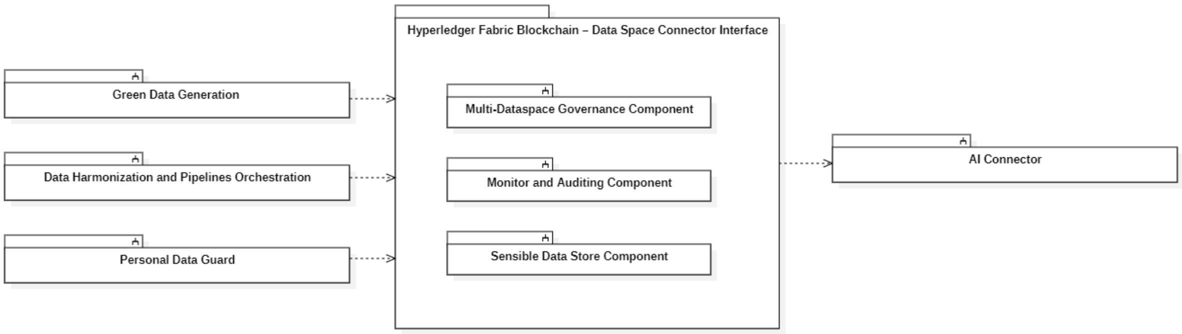

Figure 22 Hyperledger Fabric Blockchain – Data Space Connector Interface – Connection with other Components

- ***Green Data Generation (T3.1, T3.2) (From)***

The approach of the HFB interface with the IDS Connector's data plane is related to the integration of data models to be obtained from Green Data Generation node and that can be integrated into the HFB interface.

- ***Data Harmonization and Pipelines Orchestration (T3.3) (From)***

The approach of the HFB interface with the IDS Connector's data plane is related to the integration of data products to be obtained from Data Harmonization and Pipelines Orchestration node and that can be integrated into the HFB interface.

- ***Personal Data Guard (T4.7) (From)***

The HFB interface will provide the traceability and monitoring of data and transactions to the Personal Data Guard node.

- ***AI Connector (T5.6) (To)***

The HFB Interface with be connected with the AI connector that will be developed in PLIADES, as an extra layer that will provide traceability and transparency in the transactions, being a plugin to the developed connectors.

5.2.1.5 Addressed System Functional Requirements

SFR ID	Description
SFR23	Apply cryptographic techniques during data exchange
SFR24	Analyze and track data transactions
SFR26	Ensure governance and rules in the control plane
SFR28	Store sensitive data with restrictions

SFR29	Provide decentralized data connectors for secure, private, and sovereign data transfer between cloud and edge
SFR31	Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors
SFR32	Provide traceability and transparency for transactions in the entire network
SFR37	Generate data flow maps with the entire lineage of each dataset
SFR56	Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing

5.2.2 Cloud to Edge to IoT Transfer (T4.2)

5.2.2.1 Description

This tool will optimize decentralized data transfer between cloud and edge devices for IoT applications, ensuring secure, private and sovereign data exchange. It will provide novel decentralized data connectors by using IDS-RAM for security and integrating with identity management systems, ensuring compatibility with protocols and existing IDS services to increase data integrity, trust and operational efficiency within IoT networks.

5.2.2.2 Main Functionalities

- Facilitate decentralized data transfer and integration across diverse data spaces, enhancing interoperability between cloud and edge devices.
- Development of specific Decentralized Data Connectors based on strategic guidelines, allowing extensions to integrate with other edge and cloud technologies.
- Enable connectors to seamlessly integrate with existing IDS services and future technological solutions, ensuring compatibility and operational efficiency.
- Distribute data across multiple nodes, eliminating reliance on a single central authority to improve data resilience and security.

It operates on the following data types, as examples of the data that can be processed:

- ***Metadata:*** Information that provides details about other data, such as its source, creation date, format, and access permissions. Crucial for organizing and managing data effectively within IoT ecosystems.
- ***Operational Data:*** Real-time data generated by IoT devices and edge sensors, which includes metrics, such as temperature, humidity, speed, and other device-specific information.
- ***User Data:*** Personal data related to the end-users, which must be handled with strict privacy and GDPR compliance measures.

It contains the following data processes:

- ***Decentralized Data Transfer and Distribution***

Facilitation of data communication and distribution across multiple nodes without relying on a central authority, thereby improving data resilience against potential threats and reducing the risk of a single point of failure.

- ***Data Integration***

Seamless integration of data from diverse sources (cloud, edge devices, IoT sensors) into a unified system, ensuring interoperability and efficient data flow.

- ***Security and Privacy***

Utilization of IDS-RAM to assure data sovereignty and secure exchange of data using unique Decentralized Identifiers (DIDs) for each interaction between any identity and any service on the network. Use of decentralized data connectors to ensure data integrity and trust between participating entities.

- ***Data Management and Interoperability***

Management of data catalogues, identity verification, certification, and vocabularies within the data space. Ensuring connectors are compatible with existing IDS services and can integrate with future technological solutions, promoting operational efficiency and technological adaptability.

5.2.2.3 Internal Components

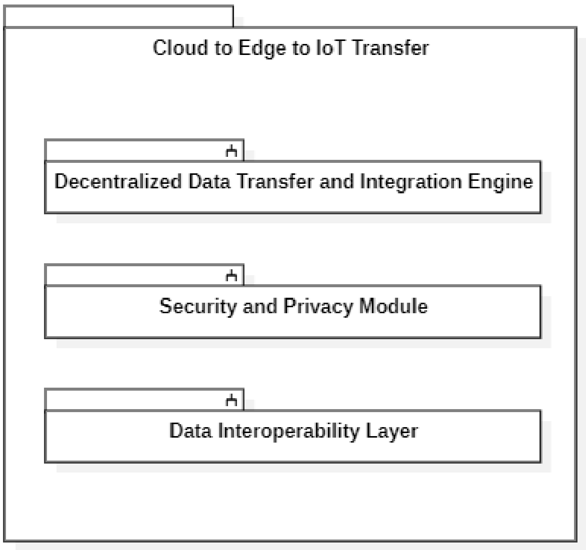

Figure 23 Cloud to Edge to IoT Transfer – Internal Components

The Cloud to Edge to IoT Transfer node consists of the following internal components:

- ***Decentralized Data Transfer and Integration Engine***

That facilitates secure and efficient data communication and integration across diverse data sources by enabling data transfer and integration thought specific connector.

- ***Security and Privacy Module***

That ensures that all data exchanges are secure and comply with data sovereignty requirements. It utilizes IDS-RAM security mechanisms to protect data during transfer and employs Decentralized Identifiers (DIDs) for identity management and integrates with T4.4 systems to verify identities and manage certificates.

- ***Data Interoperability Layer***

That manages data organization and interoperability within the IoT ecosystem.

5.2.2.4 Dependencies to other Components

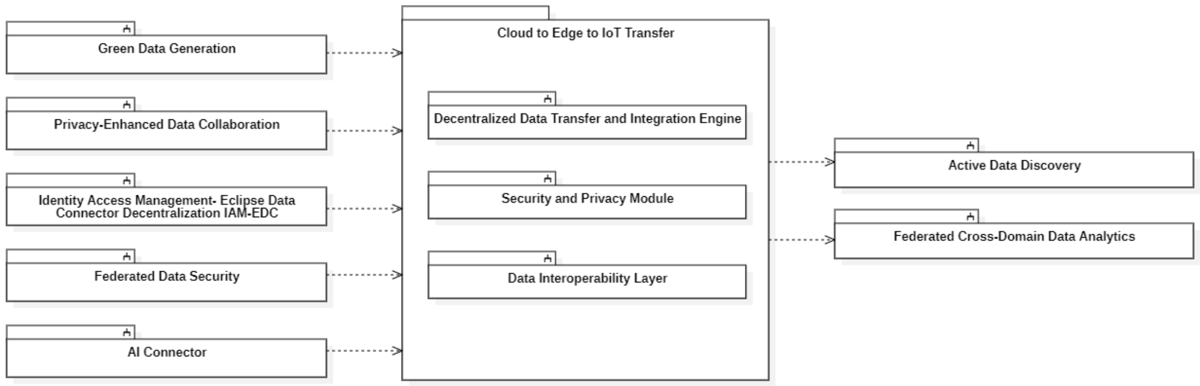

Figure 24 Cloud to Edge to IoT Transfer – Connectivity with other Components

- ***Green Data Generation (T3.1, T3.2) (From)***

Cloud to Edge to IoT transfer receives metadata, operational data, and user data from IoT sensors, edge devices, and cloud platforms.

- ***Privacy-Enhanced Data Collaboration (PEDC) (T4.3) (From)***

Cloud to Edge to IoT transfer integrates techniques and technologies developed from the Privacy-Enhanced Data Collaboration node to ensure privacy and anonymity in data sharing and collaborative intelligence processes.

- ***Identity Access Management- Eclipse Data Connector Decentralization IAM-EDC (T4.4) (From)***

Cloud to Edge to IoT transfer uses the outcomes of the Identity Access Management- Eclipse Data Connector Decentralization node to implement proper identity management systems and distributed agreement schemes.

- ***Federated Data Security (FDS) (T4.5) (From)***

Cloud to Edge to IoT transfer is based on the outcomes of Federated Data Security node to guarantee security in the data transfer processes.

- ***AI Connector (T5.6) (From)***

Cloud to Edge to IoT transfer uses the connectors developed by the AI Connector node to support data integration.

- ***Active Data Discovery (T5.4) (To)***

Cloud to Edge to IoT transfer transfers integrated and normalized data across multiple nodes ensuring secure and efficient processing, through the broker that will be created by the Active Data Discovery node, in order to be provided to all the nodes who develop AI methods.

- ***Federated Cross-Domain Data Analytics (T6.7) (To)***

Cloud to Edge to IoT transfer transfers decentralized data to the Federated cross-domain data analytics node, to extract analytics in a federated manner across decentralized data sources.

5.2.2.5 Addressed System Functional Requirements

SFR ID	Description

SFR24	Analyze and track data transactions
SFR29	Provide decentralized data connectors for secure, private, and sovereign data transfer between cloud and edge
SFR30	Provide decentralized mechanisms for identity access management
SFR31	Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors

5.2.3 Privacy-Enhanced Data Collaboration (PEDC) (T4.3)

5.2.3.1 Description

This module facilitates secure and privacy-preserving data sharing and analysis for collaborative intelligence in mobility and autonomous vehicles data spaces. It anonymizes or pseudonymizes data while enabling collaborative analysis to improve traffic management, safety, and other mobility applications.

5.2.3.2 Main Functionalities

Securely anonymize/encrypt vehicle/user data while preserving synchronized timestamps and locations. Enable privacy-preserving collaborative analysis on anonymized data for mobility intelligence. It operates on the following data types:

- Sensor data
- Location data
- User preferences

5.2.3.3 Internal Components

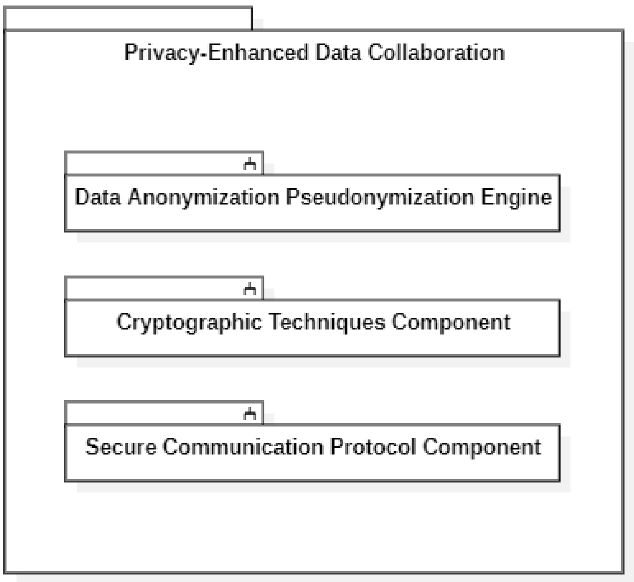

Figure 25 Privacy-Enhanced Data Collaboration (PEDC) – Internal Components

The node consists of the following internal components:

- ***Data Anonymization/ Pseudonymization Engine***

Applies techniques to remove or replace personally identifiable information (PII) from data.

- ***Cryptographic Techniques Component***

This component includes various techniques for secure data processing and analysis.

- ***Secure Communication Protocol Component***

Ensures secure data exchange between participants.

5.2.3.4 Dependencies to other Components

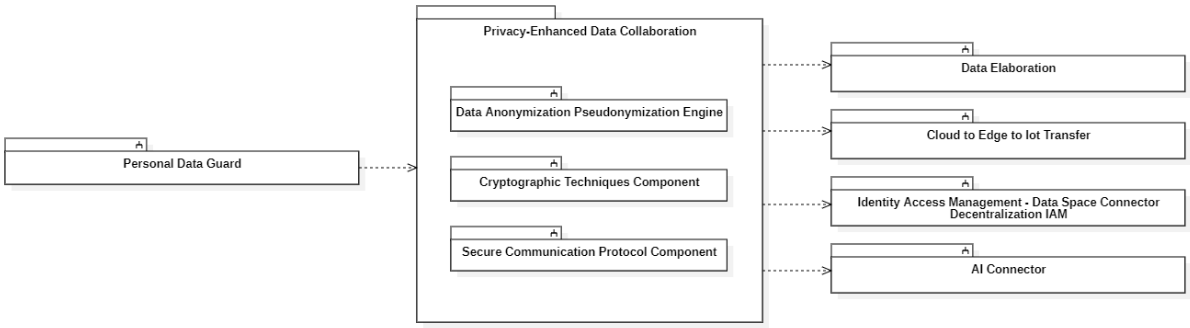

Figure 26 Privacy-Enhanced Data Collaboration (PEDC) – Connection with other Components

- ***Personal Data Guard (T4.7) (From)***

Privacy-Enhanced Data Collaboration node (PEDC) uses the outcomes from the Personal Data Guard and GDPR analysis in order to perform the privacy-preserving techniques for the data anonymization.

- ***Data Elaboration (T3.5) (To)***

Privacy-Enhanced Data Collaboration node (PEDC) provides anonymized data to the Data Elaboration node to ensure privacy-preserving techniques are ensured during data re-use.

- ***Cloud to Edge to Iot Transfer (T4.2) (To)***

Privacy-Enhanced Data Collaboration node (PEDC) provides techniques and technologies to the Cloud to Edge to IoT transfer node, to ensure privacy and anonymity in data sharing.

- ***Identity Access Management- Data Space Connector Decentralization IAM (T4.4) (To)***

Privacy-Enhanced Data Collaboration node (PEDC) provides the privacy preserving-related analysis to the IAM node, in order to ensure privacy and transparency in identity management.

- ***AI Connector (T5.6) (To)***

Privacy-Enhanced Data Collaboration node (PEDC) provides techniques for secure and anonymized data exchange through the AI connectors.

5.2.3.5 Addressed System Functional Requirements

SFR ID	Description
SFR23	Apply cryptographic techniques during data exchange
SFR25	Anonymize or pseudonymize data by removing personally identifiable information

SFR30	Provide decentralized mechanisms for identity access management
SFR33	Identify security vulnerabilities in data space architecture and protocols

5.2.4 Identity Access Management- Data Space Connector Decentralization (T4.4)

5.2.4.1 Description

This node enables the integration of permissioned blockchain environments to ensure traceability and transparency related to identity management across the value chain. In particular, it creates an abstract layer to allow decentralization mechanisms for identity while maintaining the existing capabilities of connectors with third party federated environments, without the necessity to alter their overall functionality. The software node is based on Hyperledger Fabric Blockchain in order to support fine-grained privacy and efficient queries when providing traceability and transparency.

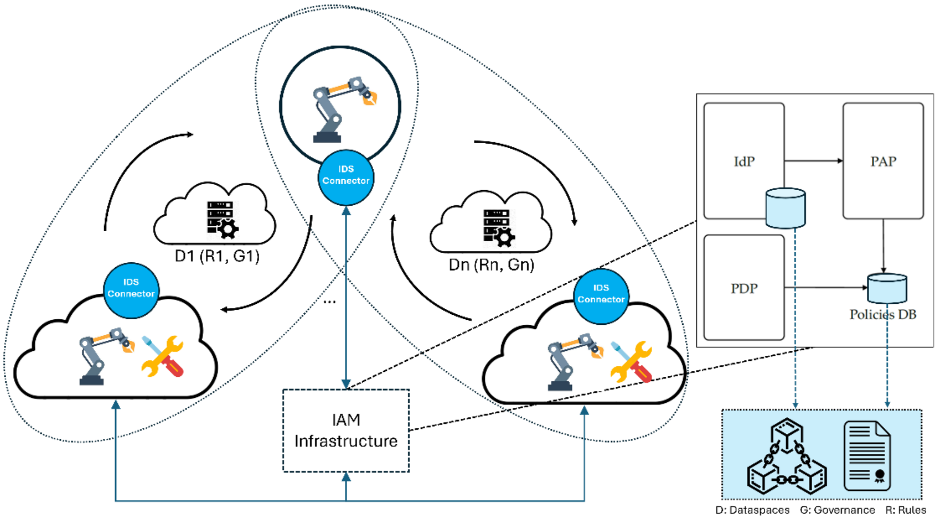

Figure 27 Decentralized IAM in data spaces

5.2.4.2 Main Functionalities

The IAM-Connector decentralization node provides traceability and transparency across the entire value chain based on a permissioned blockchain.

5.2.4.3 Internal Components

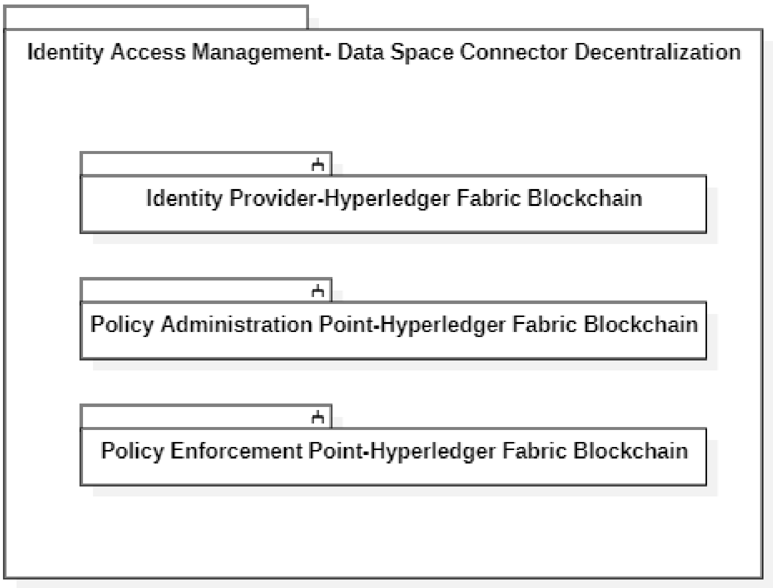

Figure 28 Identity Access Management- Data Space Connector Decentralization – Internal Components

The IAM-Connector decentralization node consists of the following internal components:

- ***Identity Provider-Hyperledger Fabric Blockchain (IDP-HFB)***

  The IDP-HFB component modifies traditional Open Source IdP Keyrock or Keycloak to include an API based interface for decentralization purposes.

- ***Policy Administration Point-Hyperledger Fabric Blockchain (PAP-HFB)***

  The PAP-HFB component modifies traditional Open-Source PAP Authzforce to include an API based interface for decentralization purposes of the Administration policies.

- ***Policy Enforcement Point-Hyperledger Fabric Blockchain (PEP-HFB)***

  The PEP-HFB component modifies traditional Open-Source PEP Wilma to include an API based interface for decentralization purposes of the enforcement policies.

5.2.4.4 Dependencies to other Components

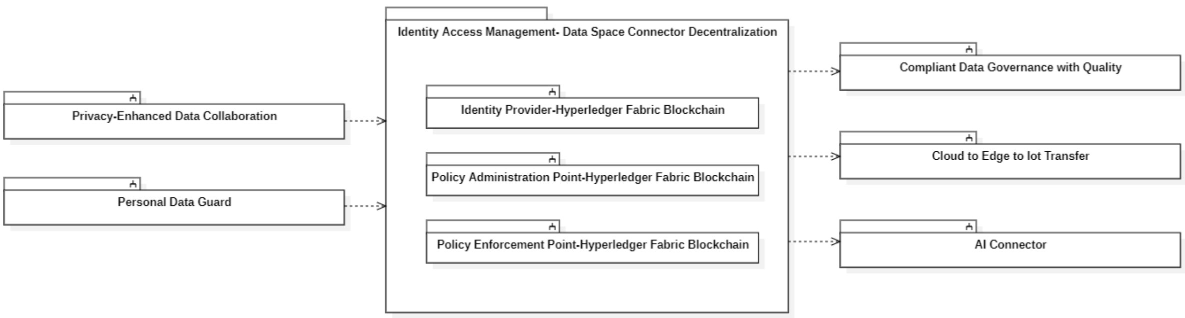

Figure 29 Identity Access Management- Data Space Connector Decentralization – Connection with other Components

- ***Privacy-Enhanced Data Collaboration (PEDC) (T4.3) (From)***

  The IAM node will use the privacy preserving-related analysis performed by the PEDC node, in order to facilitate the efficient identity management and transparency in transactions.

- ***Personal Data Guard (T4.7) (From)***

The IAM node will use the personal data identification and data lineage from the Personal Data Guard node, as part of the identity management modules that will be developed.

- ***Compliant Data Governance with Quality (CompliantDQ) (T3.7) (To)***

The IAM node will provide the developed policy engine and the policies to the CompliantDQ node, supporting the regulatory compliance component.

- ***Cloud to Edge to Iot Transfer (T4.2) (To)***

IAM decentralization node provides its outcomes to the Cloud to Edge to IoT transfer node for the development of proper identity management systems and distributed agreement schemes.

- ***AI Connector (T5.6) (To)***

The IAM node will provide the decentralized mechanisms for identity management to the AI Connector node.

5.2.4.5 Addressed System Functional Requirements

SFR ID	Description
SFR24	Analyze and track data transactions
SFR30	Provide decentralized mechanisms for identity access management
SFR32	Provide traceability and transparency for transactions in the entire network
SFR37	Generate data flow maps with the entire lineage of each dataset

5.2.5 Federated Data Security (FDS) (T4.5)

5.2.5.1 Description

This node safeguards data security across integrated data spaces in a federated environment. It leverages threat detection with AI and collaborative learning to identify and address vulnerabilities.

5.2.5.2 Main Functionalities

Track data movement, manage security updates, and leverage decentralized decision-making for continuous security. Analyze federated data space architecture for vulnerabilities and implement AI-powered threat detection with collaborative learning. It processes the following data:

- All data shared within the federated data space environment.
- Security logs and access control information.

5.2.5.3 Internal Components

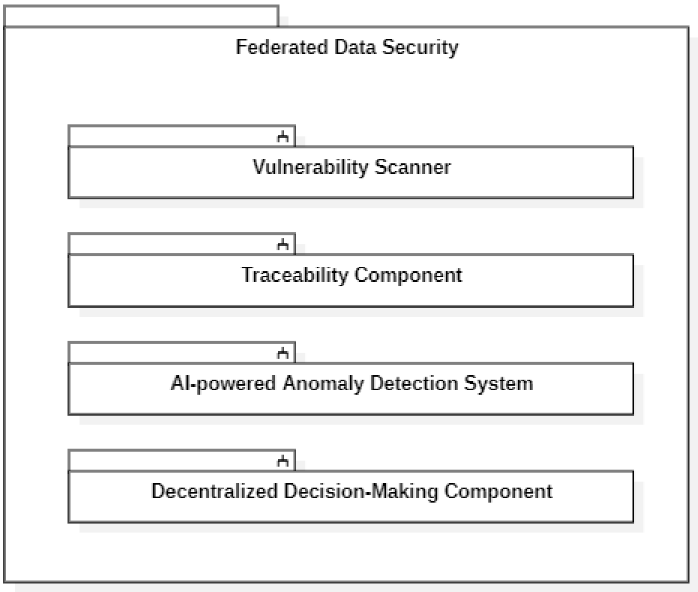

Figure 30 Federated Data Security (FDS) – Internal Components

- ***Vulnerability Scanner***

Analyzes data space architecture and protocols for security vulnerabilities.

- ***Traceability Component***

Tracks data movement and enforces traceability protocols.

- ***AI-powered Anomaly Detection System***

Identifies suspicious activity and potential security threats using ML algorithms.

- ***Decentralized Decision-Making Component***

Receives, analyzes, and prioritizes alerts from all data spaces within the federated environments. It then coordinates a response strategy and communicates it to relevant data spaces for execution.

5.2.5.4 Dependencies to other Components

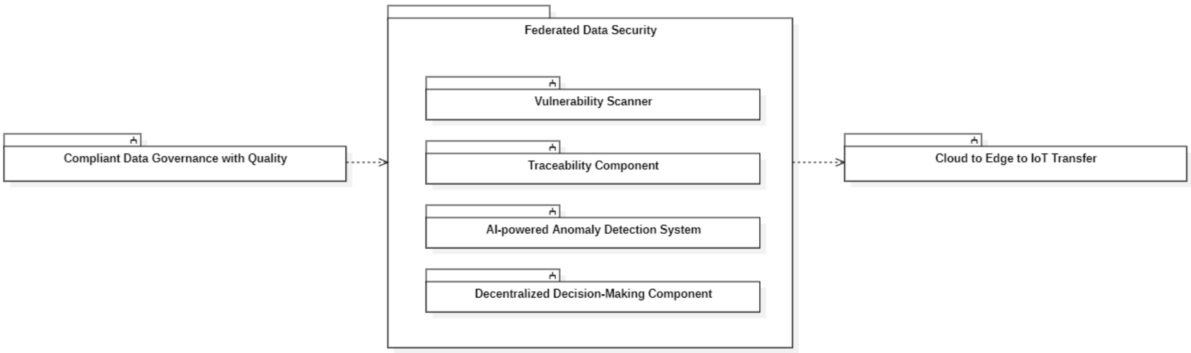

Figure 31 Federated Data Security (FDS) – Internal Components

- ***Compliant Data Governance with Quality (CompliantDQ) (T3.7) (From)***

Federated Data Security node receives data quality information from CompliantDQ node to enhance threat detection.

- ***Cloud to Edge to IoT Transfer (T4.2) (To)***

Federated Data Security node provides security analysis outcomes to the Cloud to Edge to IoT transfer node to guarantee security in the data transfer process.

5.2.5.5 Addressed System Functional Requirements

SFR ID	Description
SFR24	Analyze and track data transactions
SFR32	Provide traceability and transparency for transactions in the entire network
SFR33	Identify security vulnerabilities in data space architecture and protocols
SFR34	Identify and provide alerts for suspicious activity and potential security threads
SFR38	Identify potential privacy risks and threats

5.2.6 Relevant Data Selection (T4.6)

5.2.6.1 Description

The Relevant Data Selection node is responsible for deciding which data is worth to be stored and used for model training and retraining. The aim of this node is to measure the relevance of that data in terms of data diversity and take a decision on which data to store for obtaining a dataset more efficient in terms of the amount of data to be stored and representative of the existing variability at the same time.

5.2.6.2 Main Functionalities

Evaluation of the similarity of the incoming data to the already stored data. Filtering of the data to be stored minimizing storage and energy resources, but guaranteeing data relevance.

5.2.6.3 Internal Components

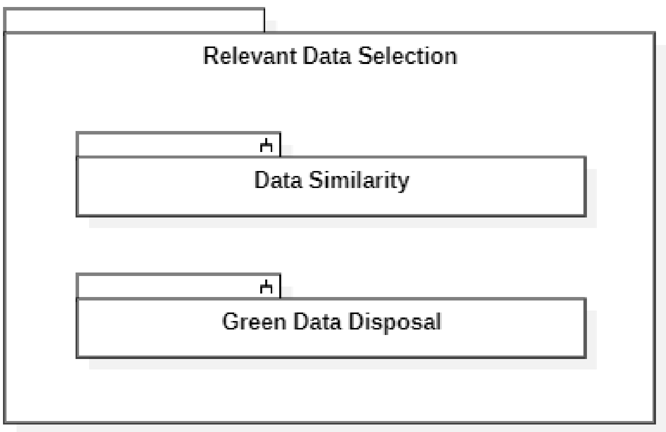

Figure 32 Relevant Data Selection – Internal Components

The Relevant Data Selection node consists of the following internal components:

- ***Data Similarity***

  Data Similarity component analyses multimodal data inputs, such as images, LiDAR, real-time analytics, etc. and measures their similarity with existing stored data in order to not save and use data which does not offer additional information for the model retraining.

- ***Green Data Disposal***

  Automatic decision-making component which uses outputs from the Data Quality node and Data Similarity component, to filter which data needs to be stored in order to preserve dataset representativity.

5.2.6.4 Dependencies to other Components

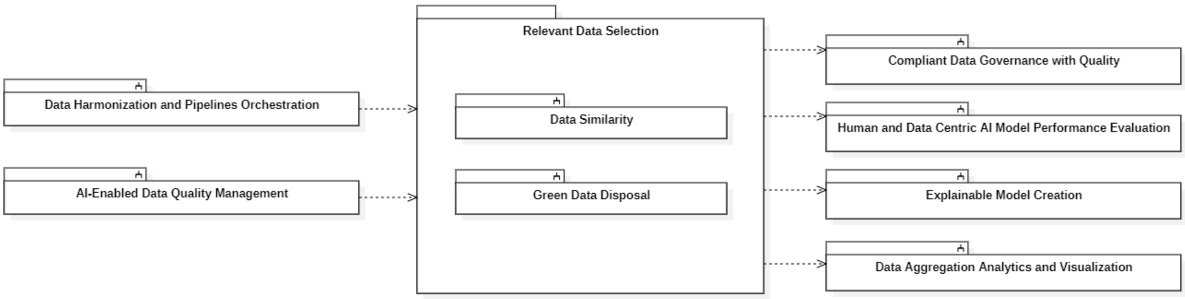

Figure 33 Relevant Data Selection – Connectivity with other Components

- ***Data Harmonization and Pipelines Orchestration (T3.3) (From)***

  The Relevant Data Selection node uses quality metrics calculated from the Data Harmonization and Pipelines Orchestration node.

- ***AI-Enabled Data Quality Management (AI-DQM) (T6.6) (From)***

  The Relevant Data Selection node uses the quality assessment performed by the AI-DQM node.

- ***Compliant Data Governance with Quality (CompliantDQ) (T3.7) (To)***

  The Relevant Data Selection node provides quality information to the CompliantDQ node.

- ***Human and Data Centric AI Model Performance Evaluation (T5.5) (To)***

  The Relevant Data Selection node provides high-quality datasets for data drift evaluation performed by the Human and Data Centric AI Model Performance Evaluation node.

- ***Explainable Model Creation (T6.1, T6.2) (To)***

  The Relevant Data Selection node provides output to the Explainable Model Creation node, so that this node uses only data of high-quality to extract explanations.

- ***Data Aggregation, Analytics and Visualization (T6.4) (To)***

  The Relevant Data Selection node provides output to the Data Aggregation, Analytics, and Visualization (DAAV) node, so that only useful data are taken into account during aggregation and analytics extraction.

5.2.6.5 Addressed System Functional Requirements

SFR ID	Description

SFR5	Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics
SFR6	Prioritize data based on their quality metrics
SFR43	Extract similarity metrics for multimodal data
SFR44	Discard data based on similarity metrics

5.2.7 Personal Data Guard (T4.7)

5.2.7.1 Description

This node is focused on managing and tracking personal data across its lifecycle to ensure it meets GDPR requirements, including identification, flow mapping, and compliant disposal. This tool will function on the connectors of the dataspace following the IDSA dataspace protocol.

5.2.7.2 Main Functionalities

- Smart personal data identification service
- Data lineage
- Data disposal on expiration data

5.2.7.3 Internal Components

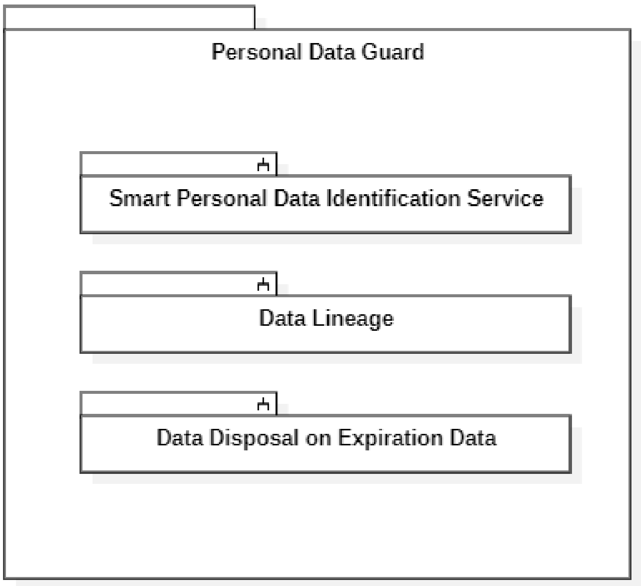

Figure 34 Personal Data – Internal Components

The Personal Data Guard module consists of the following internal components:

- ***Smart Personal Data Identification Service***

  This component is responsible for identifying and distinguishing between personal and non-personal data. It uses as input data sets from various project components containing mixed personal and non-personal data, along with metadata to help classify data types (e.g., source, ownership, contain description). It produces labelled data sets with distinctions between personal and non-personal data, metadata tags indicating the data type, sensitivity level, and classification results and an alert when unexpected personal data is detected.

- ***Data Lineage***

Maps the flow of data to monitor and track lineage, identifying where personal data is stored, transferred, and processed. It uses as input labelled data sets with identified personal data from the Smart Personal Data Identification Service, metadata on data sources, processing locations, transfer logs, and usage history. It produces a data flow map showing the entire lineage of each data set, including where it is stored, processed, and transferred, along with alerts or risk assessments related to data flow, identifying potential privacy risks and unauthorized transfers.

- ***Data Disposal on Expiration Data***

Annotates each data set with a "legal disposal date" to support compliance with data retention policies. It uses as input data sets with identified personal data from the Smart Personal Data Identification Service, metadata specifying data types, sensitivity levels, and compliance requirements, GDPR policies or rules indicating required retention periods and disposal requirements. It outputs annotated data sets with a “legal disposal date” indicating when data should be deleted and alerts for data disposal actions when the legal disposal date has passed.

5.2.7.4 Dependencies to other Components

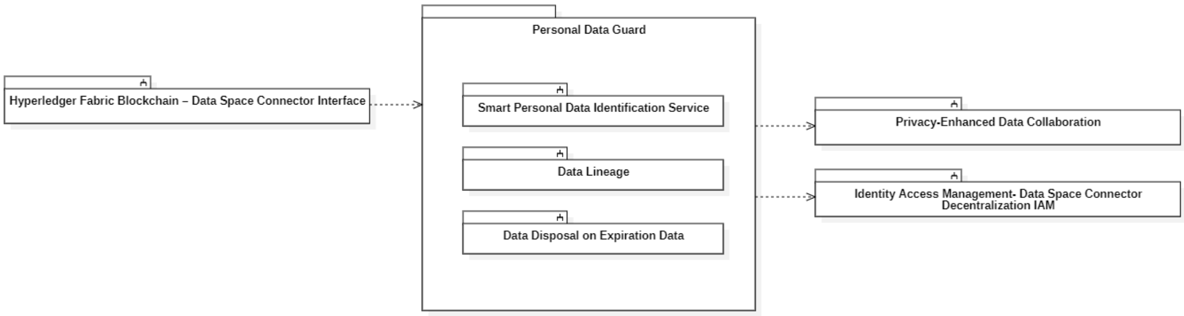

Figure 35 Personal Data – Internal Components

- ***Hyperledger Fabric Blockchain – Data Space Connector Interface (HFB Interface) (T4.1) (From)***

The Personal Data Guard uses the traceability produced by the HFB Interface to perform the data lineage and data flow mapping.

- ***Privacy-Enhanced Data Collaboration (PEDC) (T4.3) (To)***

The Personal Data Guard node provides its outcomes to the PEDC node to develop and perform privacy-preserving techniques for data anonymization.

- ***Identity Access Management- Data Space Connector Decentralization IAM (T4.4) (To)***

The Personal Data Guard node provides the personal data identification and Data lineage outcomes to the identity management components of IAM.

5.2.7.5 Addressed System Functional Requirements

SFR ID	Description
SFR36	Identify personal and non-personal data

SFR37	Generate data flow maps with the entire lineage of each dataset
SFR38	Identify potential privacy risks and threats
SFR45	Perform automated data disposal processes based on expiration date

5.3 Functional Nodes from WP5 vantage point

The functional nodes for WP5-level are presented in the following diagram (Figure 36).

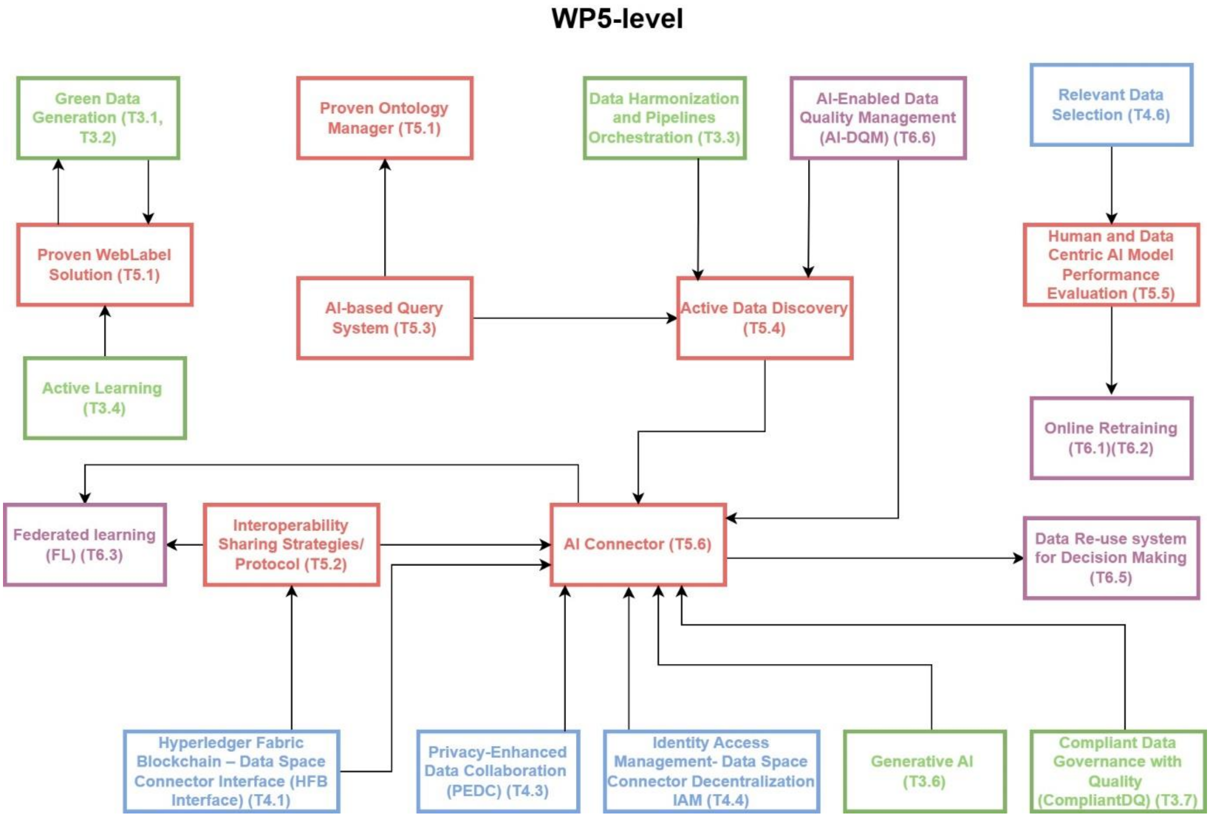

Figure 36 WP5-level Architecture

5.3.1 Proven WebLabel Solution (T5.1)

5.3.1.1 Description

This node consists of a set of tools for the annotation of multimedia content. These tools are in the form of web applications that allow the input of human knowledge in the process of generating visual annotations.

5.3.1.2 Main Functionalities

- The web applications are designed for labelling multimedia content (videos, images, pointclouds, etc).
- It has a configurable UI to enable multiple use cases.
- Designed to be orchestrated and scalable in cloud architectures.
- Compliant with ASAM OpenLabel 1.0 standard.
- The implementation of the tool functions is based on the VCD toolkit.

It operates on the following data:

- ***Input Data Types:*** Multimedia content (videos, images, point clouds, etc.), configuration files and OpenLabel pre-annotation files
- ***Output Data Types:*** OpenLabel annotation files

5.3.1.3 Internal Components

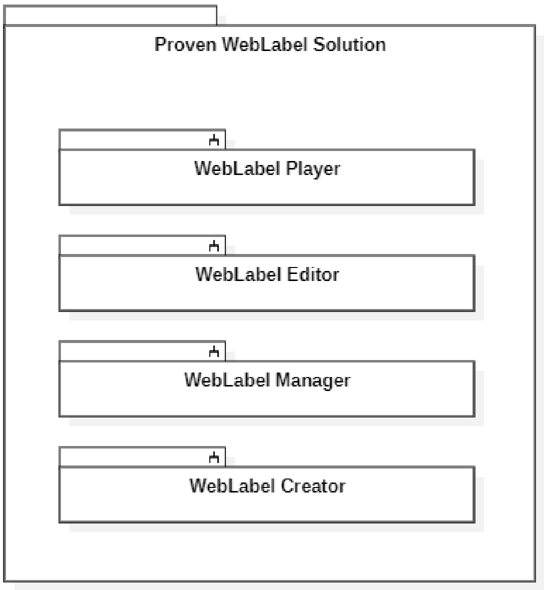

Figure 37 Proven WebLabel Solution – Internal Components

The Proven WebLabel solution consist of the following components:

- ***WebLabel Player***

This component allows the visualization of already annotated multimedia content.

- ***WebLabel Editor***

This component allows the annotation or correction of already annotated content by a human expert.

- ***WebLabel Manager***

This component allows the generation of annotation tasks and assign those tasks to the human annotator labor force. In addition, these tasks enable the triggering of semi-automatic annotation pipelines.

- ***WebLabel Creator***

This component allows the configuration and definition of the annotation tasks and generates the necessary structure configuration files required by the WebLabel Editor component.

5.3.1.4 Dependencies to other Components

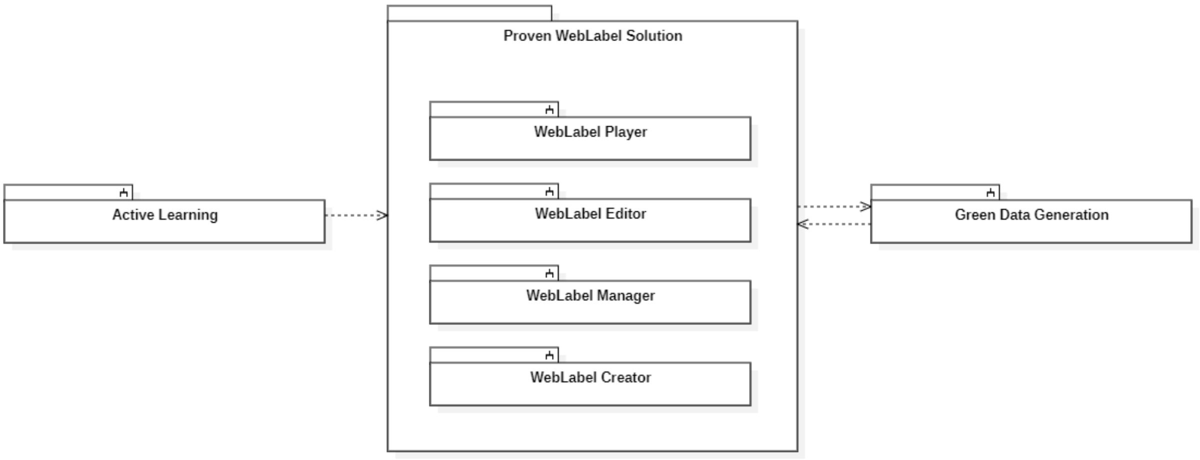

Figure 38 Proven WebLabel Solution – Connection with other Components

- ***Green Data Generation (T3.1, T3.2) (From/To)***

The Proven Weblabel Solution node provides its tools to the annotation of the datasets performed by the Green Data Generation node, in order to provide annotation efficiently using a common format. Also, uses the data generated by the Green Data Generation to modify existing labels and visualize them.

- ***Active Learning (T3.4) (From)***

The Proven Weblabel Solution node receives the techniques developed by the Active Learning node to be included in the semi-automated annotation pipelines and used by the Weblabel Solution.

5.3.1.5 Addressed System Functional Requirements

SFR ID	Description
SFR9	Generate annotation labels for data in a semi-automated manner
SFR10	Extract automated labels for datasets assigned with confidence level
SFR11	Allow human validation of dataset labels with low confidence
SFR12	Inject human knowledge into data annotation and metadata creation processes
SFR13	Annotate multimodal data
SFR14	Visualize existing annotations of multimodal data
SFR18	Interface to collect human feedback for improving AI performance

5.3.2 Proven Ontology Manager (T5.1)

5.3.2.1 Description

Proven Ontology Manager (POM) is a node that serves semantic services to other applications such as semantic labelling systems and semantic inference. POM is defined in the context of Artificial Intelligence (AI) development, and for Connected, Cooperative and Automated Mobility (CCAM) solutions.

5.3.2.2 Main Functionalities

POM shall have a reachable API, with a set of functions to cover semantic functionalities:

- Load ontology in a graph database: an ontology file or URL is provided to POM, which loads it.
- Get ontology: POM returns ontology files
- Get/find class: POM returns information about a certain class: description, etc.
- Get/find class relations: POM returns information about all relations of a class.
- Check class or relation validity.
- Load data compliant with ontology data model (i.e., OpenLabel)
- Creation of SPARQL queries to find specific content.
- Dashboard to explore database

It operates on the following data types:

- ***Input Data Types***: OpenLabel schema compliant ontology and dataset files.
- ***Output Data Types***: OpenLabel query results.

5.3.2.3 Internal Components

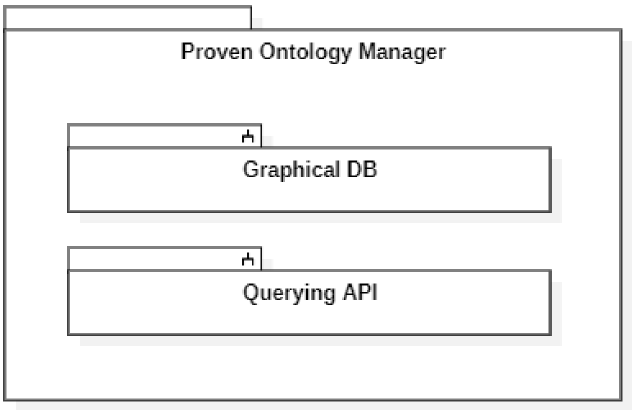

Figure 39 Proven Ontology Manager – Internal Components

The Proven Ontology Manager consist of the following components:

- ***Graphical DB***

  This component consists of a database based on GraphDB which will contain the ontology and the data of the specific use case.

- ***Querying API***

  This component consists of an API which allows making queries to the database using a common format, such as SparQL. These queries allow the exploration of the entire dataset and find specific complex situations by leveraging the semantic context and relations of the data.

5.3.2.4 Dependencies to other Components

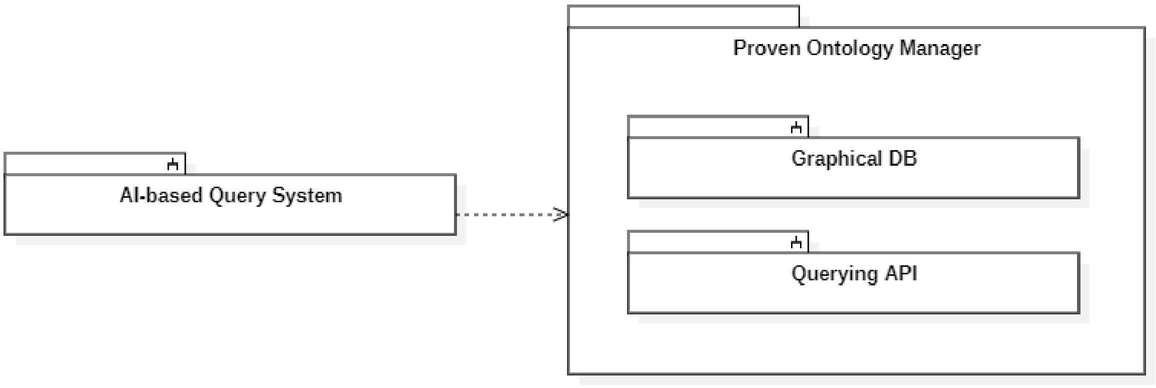

Figure 40 Proven Ontology Manager – Connection to other Components

- ***AI-based Query System (T5.3) (From)***

The Proven Ontology Manager node will receive from the AI-based Query System the queries in an understandable format, after being transformed from high-level human language, allowing the interpretation of user intent to discover and access data.

5.3.2.5 Addressed System Functional Requirements

SFR ID	Description
SFR46	Allow data exchange with ontology-based components
SFR48	Store the ontology and data into a structured database
SFR49	Allow queries to explore existing databases and internal relations
SFR52	Provide optimal recommendations between datasets and data consumer needs

5.3.3 Interoperability Sharing Strategies/ Protocol (T5.2)

5.3.3.1 Description

The Interoperability Sharing strategies node is responsible for the interoperable data sharing between entities within and across Data Spaces, taking into account separated data spaces (individual data spaces as in the case of PLIADES that may have been constituted based on the activity sector), overlapping data spaces or nested data spaces (as it can be the industrial data spaces and for example advanced manufacturing dataspace, that may have some strategies in common). The Data Sharing strategies for interoperability node is based on the latest Data Spaces Blueprint V1.5 as well as the insights of the Eclipse Dataspace Protocol. The aim of this node is to provide solid and standardized strategies that will ensure data interoperability between data spaces that integrate and connect diverse types of data in a coherent way, allowing to be accessed and shared by different applications and users.

5.3.3.2 Main Functionalities

Expose the Dataspaces Protocol version that each connector has implemented in order to allow the seamless operation between connectors taking part in the data exchange.

- ***Data Processes***

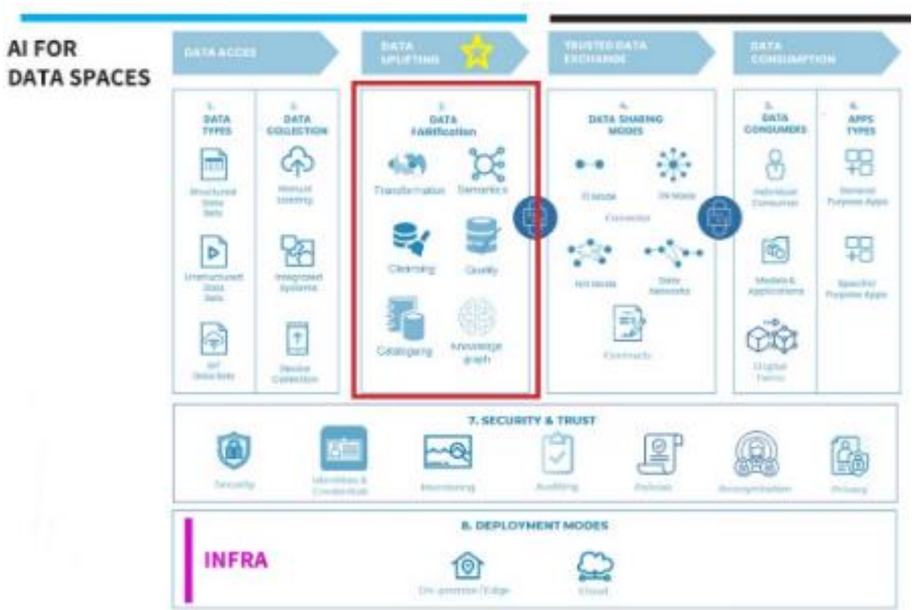

Figure 41 Interoperability strategies as part of the Data uplifting phase of the Data exchange

As seen in the figure above, the strategies for interoperability should work with any data type collected from the data provider and requested from the data consumer in all data spaces identified in the PLIADES project. It will include also semantic interoperability, but goes beyond this area to allow a technical connection between data by the standardization of datasets, data catalogs, APIs etc.

5.3.3.3 Internal Components

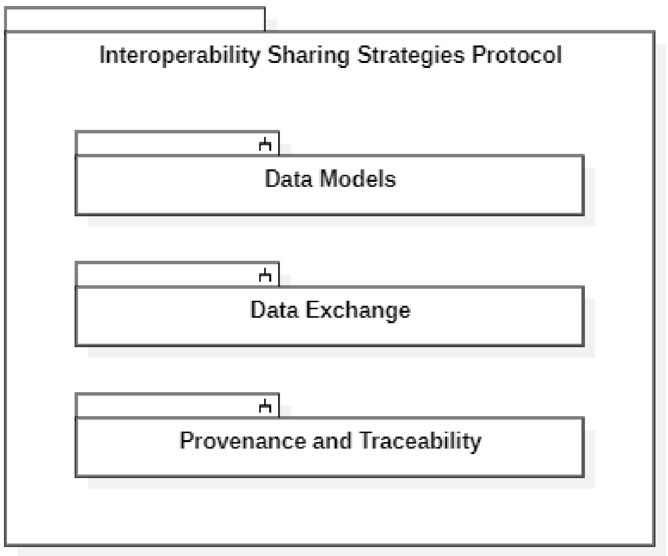

Figure 42 Interoperability Sharing Strategies/ Protocol – Internal Components

The Data Sharing strategies for interoperability node consists of the following internal components:

- ***Data Models***

Data models ensure that data is accurately and consistently interpreted when exchanged within a data space. The data model consists of metadata that provides information about semantics, helping to interpret the actual exchanged data. Data spaces should consider shared data models, or 'semantic standards' and if existing models do not suffice, explore the ability to extend an existing model. Only if this is not viable, a data space should create a new model reusing existing concepts from existing standards where possible.

The following capabilities are the ones that this component will be focused on harmonizing:

  - ***Data Model Development****:* Develop and reuse of data models to ensure uniformity and interoperability.
  - ***Data Model Abstraction:*** Transition from semantic to technical interoperability, enabling the representation of the exchanged data (see also: Data Exchange building block).
  - ***Data Model Governance:*** Governance and management aspect of data models, including tools and processes to main data models to ensure wide consensus regarding the data model used in the data space.
  - ***Data Models Across Data Spaces:*** Discovery of data models across data spaces, enabling multiple data spaces to become semantic interoperable.

- ***Data Exchange***

The scope of this component is the transmission of the data between participants on a data space, by achieving agreed mechanisms between participants (for the actual transmission of data between participants), for which data spaces need to make a choice. The data space has to be capable of maintaining a precise inventory of the technical specifications for the different transfer/access protocols used and their different versions.

To exchange data, the data space participants must use a common protocol, which includes the syntax and the sequence of the interaction. This protocol is used to define an API. While APIs may vary across domains, standardization within a data space is needed to enable interoperability.

In domains lacking a consensus protocol, a protocol accepted by most participants, proposing a data exchange protocol that is *agnostic to the data structure* is crucial. Depending on the scope of the respective data space and coverage of concrete use cases, the data exchange protocol should support the following functionalities:

  - Efficient transmission of data
  - Publication of the protocol specification in the data space that enables the access to the data asset

- ***Provenance and Traceability***

While dealing with cross-sectorial data spaces, it may happen that certain use cases need metadata over the actual data that is being shared in order to allow auditing and compliance. This component provides guidance to support provenance, traceability, logging and audits in a standardized way.

5.3.3.4 Dependencies to other Components

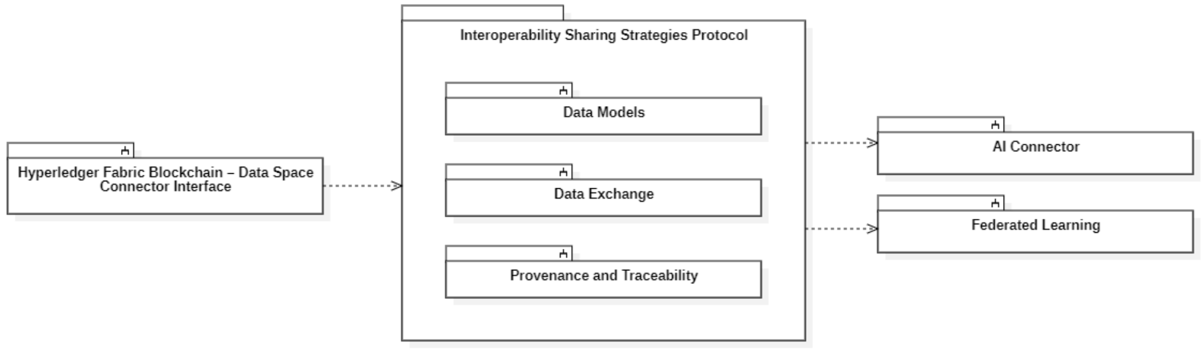

Figure 43 Interoperability Sharing Strategies/ Protocol – Connectivity with other Components

In general terms, this node is strongly related to components aimed at Data exchange/transmission, such as connectors, as well as components related to tracking the data sharing process itself for auditing and compliance purposes.

- ***Hyperledger Fabric Blockchain – Data Space Connector Interface (HFB Interface) (T4.1) (From)***

The Interoperability sharing strategies receive input from the HFB Interface regarding the tracking of the data sharing processes.

- ***AI Connector (T5.6) (To)***

The interoperability sharing strategies provide output to the AI connector node developed in PLIADES, allowing the interoperability between data spaces.

- ***Federated Learning (FL) (T6.3) (To)***

The interoperability sharing strategies developed will be provided to the Federated Learning node, as part of the heterogeneous data sources and different data lakes that will be used.

5.3.3.5 Addressed System Functional Requirements

SFR ID	Description
SFR40	Enable cross-domain data discovery based on context and predefined criteria
SFR47	Perform cross-domain adaptation between datasets
SFR50	Provide strategies for data interoperability between data spaces

5.3.4 AI-based Query System (T5.3)

5.3.4.1 Description

This node incorporates novel conversational LLM models to the process of performing queries upon a dataset. Through the use of context-aware NLP mechanisms, knowledge graphs and data harmonization, the system will transform this understandable language into a complex machine understandable language.

5.3.4.2 Main Functionalities

The system shall have the following functionalities:

- Ingest high-level domain specific language.
- Convert this language into a valid query capable to be ingested by semantic data discoverable nodes.
- Perform queries upon Graphical Databases.

It operates on the following data:

- ***Input Data Types***: Human high-level domain specific language queries.
- ***Output Data Types***: OpenLabel query results.

5.3.4.3 Internal Components

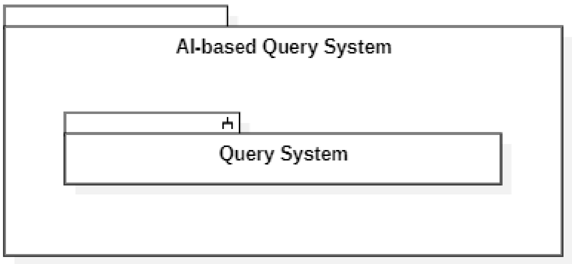

Figure 44 AI-based Query System – Internal Components

The AI-based Query System consist of the following component:

- ***Query System***

This component contains the overall functionality of the module, which is responsible to receive the human language and transform it into machine language through NLP mechanisms.

5.3.4.4 Dependencies to other Components

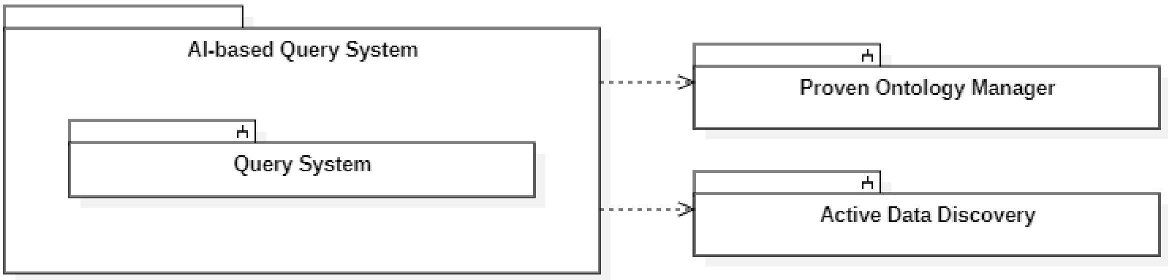

Figure 45 AI-based Query System – Connections with Internal Components

- ***Proven Ontology Manager (T5.1) (To)***

The AI-based Query System will transform the high-level human language into a query format understandable by the Proven Ontology Manager, which will allow the interpretation of user intent to discover and access data.

- ***Active Data Discovery (T5.4) (To)***

The AI-based Query System will provide the queries processing methods to the Active Data Discovery node.

5.3.4.5 Addressed System Functional Requirements

SFR ID	Description
SFR49	Allow queries to explore existing databases and internal relations
SFR51	Receive natural language inputs to perform queries in Graphical Databases
SFR52	Provide optimal recommendations between datasets and data consumer needs

5.3.5 Active Data Discovery (T5.4)

5.3.5.1 Description

The Active Data Discovery node will enable data discovery and interoperability through an AI-boosted broker. The main goal is for the broker to be enhanced with cross-domain capabilities. The node will be able to communicate and connect with domain metadata brokers corresponding to the five domains of the PLIADES framework and perform an optimized matching between available datasets, data providers, and data consumers of the ecosystem. The AI-brokers' added functionalities will be ensured with the use of AI services.

5.3.5.2 Main Functionalities

Based on the above, the Active Data Discovery node will present the following functionalities:

- Recommend the optimal matching between datasets and data consumers.
- Facilitate cross-domain data discovery.
- Integrate recommendation engines, ranking systems, and AI-enabled declarative querying.

Data Processes

Homogenized aggregated metadata trained in a federated way, received from the available brokers of each data space. The metadata will be produced by analytics techniques developed in WP6. The data processing within the AI-broker may include:

- Metadata handling
- Analyzing the data through the AI components
- Normalizing and combining the cross-domain data

5.3.5.3 Internal Components

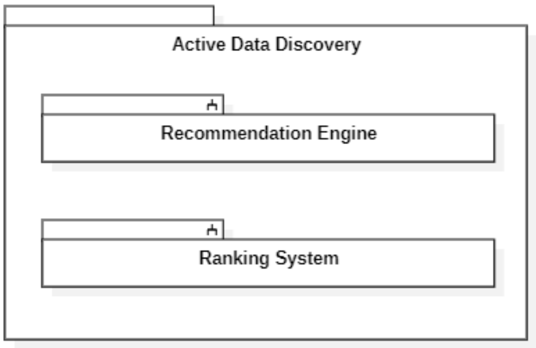

Figure 46 Active Data Discovery – Internal Components

- ***Recommendation Engine***

The recommendation engine will analyze metadata/ontologies or other kinds of references from interconnected data spaces to identify and suggest relevant datasets, services, or partners based on users' contextual needs and historical interactions.

- ***Ranking System***

The ranking system will prioritize datasets, services, or partners by evaluating relevance, quality, and trustworthiness based on predefined criteria and user context. It operates by analyzing references such as metadata, usage patterns, and compliance scores to deliver the most pertinent results while adhering to IDSA principles of transparency and fairness.

5.3.5.4 Dependencies on other Components

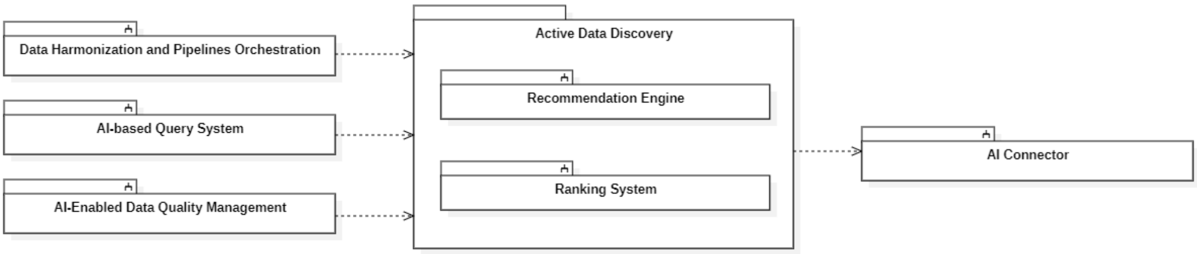

Figure 47 Active Data Discovery – Connection with other Components

- ***Data Harmonization and Pipelines Orchestration (T3.3) (From)***

The AI-boosted broker will receive data products from the Data Harmonization and Pipelines Orchestration node and use this information to provide suggestions from its encapsulated services (Ranking System, Recommendation Engine, declarative querying system).

- ***AI-based Query System (T5.3) (From)***

The Active Data Discovery node will use the queries processing methods incorporated in the AI-based Query System to support the Declarative querying component.

- ***AI-Enabled Data Quality Management (AI-DQM) (T6.6) (From)***

The Active Data Discovery node will use quality metrics and the improved data from the AI-DQM node, to take into account the quality during the extraction of recommendations.

- ***AI Connector (T5.6) (To)***

The output will be a suggested matching of the data consumer and data provider, which will be provided to the AI connector node, so that the data transactions will be performed.

5.3.5.5 Addressed System Functional Requirements

SFR ID	Description
SFR6	Prioritize data based on their quality metrics
SFR40	Enable cross-domain data discovery based on context and predefined criteria
SFR52	Provide optimal recommendations between datasets and data consumer needs
SFR53	Evaluate continuously the quality of data
SFR55	Extract decentralized analytics in real-time

5.3.6 Human and Data Centric AI Model Performance Evaluation (T5.5)

5.3.6.1 Description

The Human and Data Centric AI Model Performance Evaluation node is responsible for alerting the end-user of possible incoming data drifts and/or for collecting feedback on model performance. Its main purpose is to collect in-production corrections of model predictions and other relevant end-user feedback. This allows to detect possible retraining needs and to select relevant data.

5.3.6.2 Main Functionalities

Measurement of the data drift, drift alert notification and collection of end-user feedback on model performance.

5.3.6.3 Internal Components

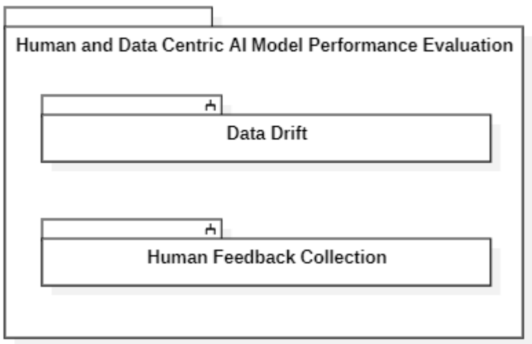

Figure 48 Human and Data Centric AI Model Performance Evaluation – Internal Components

The Human and Data Centric AI Model Performance Evaluation node consists of the following internal components:

- ***Data Drift***

Data Drift component receives as input multimodal data, such as images, LiDAR, real-time analytics, etc., compares distributions of existing data and new incoming data and notifies alerts when data distributions are too different.

- ***Human Feedback Collection***

The Human Feedback component provides the end-user with an interface to perform actions, such as selecting relevant incoming datasets to improve the model (due to poor model performance or other reasons), correcting model predictions for specific samples or providing other useful feedback for model retraining.

5.3.6.4 Dependencies to other Components

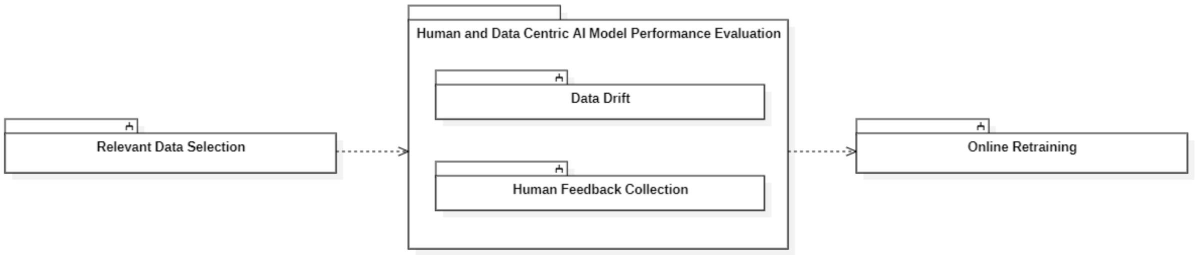

Figure 49 Human and Data Centric AI Model Performance Evaluation – Connection with other Components

- ***Relevant Data Selection (T4.6) (From)***

The Human and Data Centric AI Model Performance Evaluation uses as input datasets from the Relevant Data Selection node to perform data drift evaluation.

- ***Online Retraining (T6.1, T6.2) (To)***

The human and data centric AI model performance evaluation node provides indication on re-training needs to the Online Retraining node.

5.3.6.5 Addressed System Functional Requirements

SFR ID	Description
SFR17	Establish automated monitoring and evaluation mechanisms for tracking AI model performance improvements
SFR18	Interface to collect human feedback for improving AI performance
SFR22	Develop online re-training strategies for AI models
SFR41	Identify data drifts on multimodal data and provide alerts when detected

5.3.7 AI Connector (T5.6)

5.3.7.1 Description

The AI Connector node will be developed to establish data transformation, security, and privacy-preserving data sharing. It will comply with the IDS standards and will expand the capabilities of the IDS Connector. To achieve this, communication with other cross-domain services of WP3, WP4 and

WP5 will be performed. The establishment of the foreseen communication with these services will add enhanced functionalities to the connector, allowing it to perform cross-domain tasks.

5.3.7.2 Main Functionalities

The node will be in the position to assist with active data discovery, active data governance, virtual replica, interoperability annotations, data quality assessment, and identity provider and to extend the data spaces interoperability. So, the main functionalities can sum up to:

- Enable seamless operation across multiple data spaces.
- Support decentralized data transformation and analysis.
- Ensure privacy-preserving and secure data sharing.

The connector will be in the position to process data and analyze metadata produced in WP6, alongside related data usage policies stored in each data space. It operates on the following data types: Multimodal data including text, images, and metadata from various domains.

5.3.7.3 Internal Components

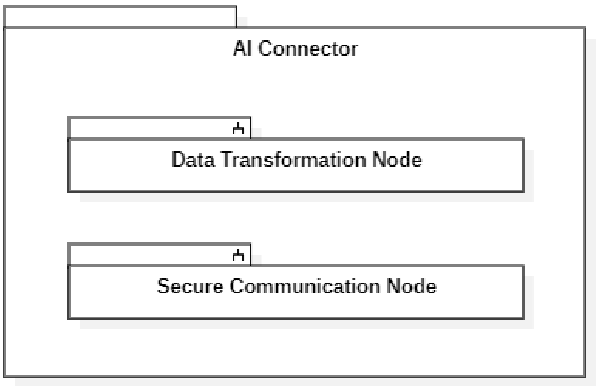

Figure 50 AI Connector – Internal Components

As the connectors will serve many purposes and interact with multiple components the strict architecture cannot be specified at this early stage. It is highly possible to integrate with components of WP3. Some abstract internal components are:

- ***Data Transformation Node***

  Manages the transformation of data between different formats and standards to ensure interoperability across domains.

- ***Secure Communication Node***

  Ensures secure and privacy-preserving data exchange by implementing encryption and data masking techniques.

5.3.7.4 Dependencies on other Components

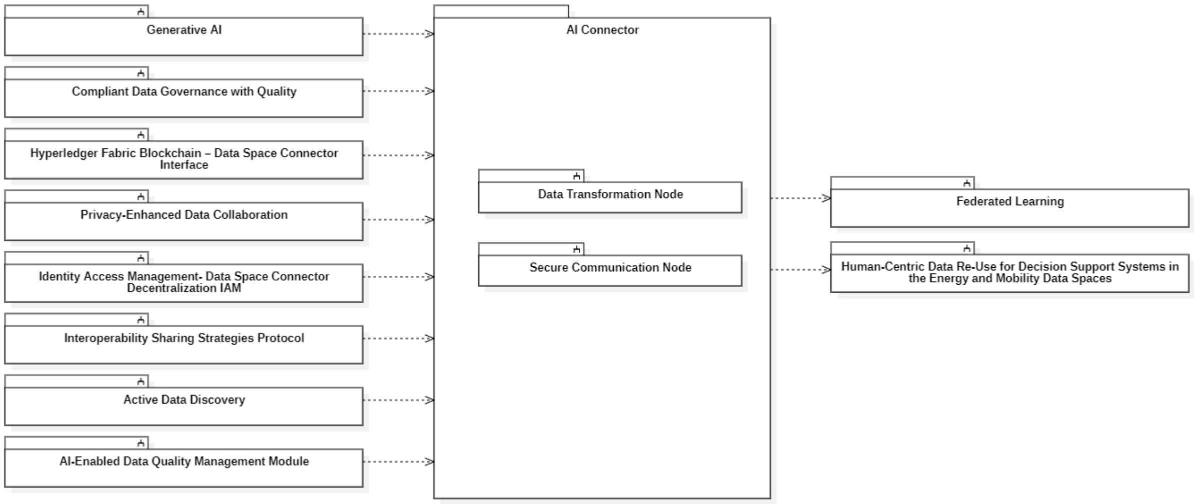

Figure 51 AI Connector – Connection with other Components

The AI Connector is expected to communicate via special nodes (will be developed to be compatible with IDS standards) with the following nodes:

- ***Generative AI (T3.6) (From)***

  The AI Connector will receive artificial data from the Generative AI node, providing access to more data of any condition/domain needed.

- ***Compliant Data Governance with Quality (CompliantDQ) (T3.7) (From)***

  The AI Connector will use the data governance practices developed in CompliantDQ node, as part of the connectors that will be developed.

- ***Hyperledger Fabric Blockchain – Data Space Connector Interface (HFB Interface) (T4.1) (From)***

  The AI Connector will use the HFB Interface as a plugin to monitor the transactions.

- ***Privacy-Enhanced Data Collaboration (PEDC) (T4.3) (From)***

  The AI Connector received the secure and privacy-preserving data sharing and analysis techniques developed in the PEDC node.

- ***Identity Access Management- Data Space Connector Decentralization IAM (T4.4) (From)***

  The AI Connector will use the IAM node to ensure privacy and support efficient queries with traceability and transparency.

- ***Interoperability Sharing Strategies/ Protocol (T5.2) (From)***

  The AI Connector will receive the interoperability sharing strategies by the Interoperability Sharing Strategies node, allowing the interoperability between data spaces.

- ***Active Data Discovery (T5.4) (From)***

  The AI Connector will receive the suggested matching of the data consumer and data provider performed by the Active Data Discovery node.

- ***AI-Enabled Data Quality Management (AI-DQM) Node (T6.6) (From)***

The AI Connector will use the automated data quality analysis methods by the AI-DQM node, to utilize during the data transactions.

- ***Federated Learning (FL) (T6.3) (To)***

Federated learning will use the AI-based connectors for data transformation, secure and privacy-preserving data sharing.

- ***Data Re-use System for Decision Support Systems (T6.5) (To)***

The developed AI Connectors will be used as a basis for the development of extended connectors for data reuse in energy and mobility by the Data Re-use System for Decision Support Systems node.

5.3.7.5 Addressed System Functional Requirements

SFR ID	Description
SFR29	Provide decentralized data connectors for secure, private, and sovereign data transfer between cloud and edge
SFR31	Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors
SFR32	Provide traceability and transparency for transactions in the entire network
SFR46	Allow data exchange with ontology-based components
SFR56	Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing

5.4 Functional Nodes from WP6 vantage point

The functional nodes for WP6-level are presented in the following diagram (Figure 52).

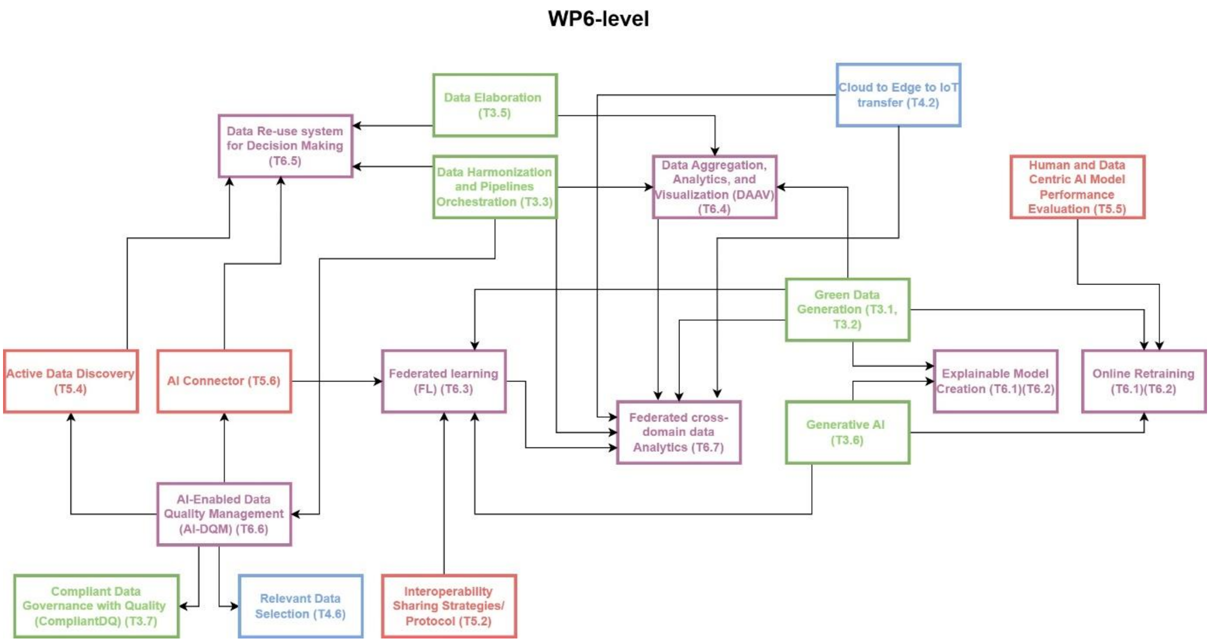

Figure 52 WP6-level Architecture

5.4.1 Explainable Model Creation (T6.1, T6.2)

5.4.1.1 Description

The Explainable Model Creation node is responsible for the development of explainable AI models that will provide reasonable justification for their decisions.

5.4.1.2 Main Functionalities

Develop explainable AI/ML models.

5.4.1.3 Internal Components

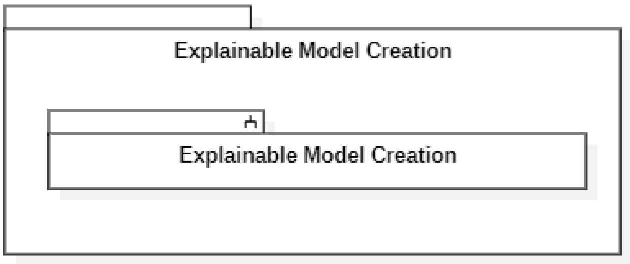

Figure 53 Explainable Model Creation – Internal Components

The Explainable Model Creation node consists of the following internal component:

- ***Explainable Model Creation***

  Explainable Model Creation module uses as input multimodal data, such as images, LiDAR, real-time analytics, etc., real or artificial, in order to develop and validate explainable AI models.

5.4.1.4 Dependencies to other Components

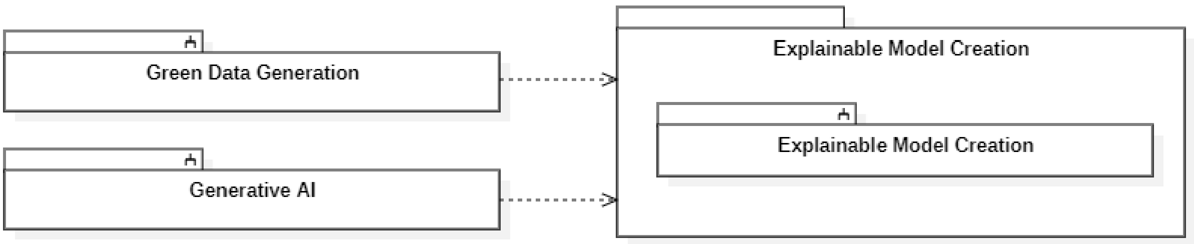

Figure 54 Explainable Model Creation – Connectivity with other Components

Explainable Model Creation node interacts with the following components:

- ***Green Data Generation (T3.1, T3.2) (From)***

  The Explainable Model Creation node receives data from the Green Data Generation, required to train explainable AI methods.

- ***Generative AI (T3.6) (From)***

  The Explainable Model Creation node uses data from the Generative AI node, required to train explainable AI methods.

5.4.1.5 Addressed System Functional Requirements

SFR ID	Description
SFR21	Provide explanations to AI models' inference

5.4.2 Online Retraining (T6.1, T6.2)

5.4.2.1 Description

The Online Retraining node is responsible for the development of strategies for retraining AI models on-the-fly, in order to adapt in changing scenarios with increased performance.

5.4.2.2 Main Functionalities

Develop online re-training strategies.

5.4.2.3 Internal Components

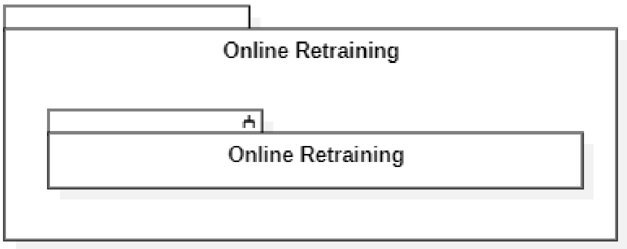

Figure 55 Online Retraining – Internal Components

The Online Retraining node consists of the following internal component:

- ***Online Retraining***

  Online Retraining node uses as input multimodal data, such as images, LiDAR, real-time analytics, etc., real or artificial, in order to develop online re-training strategies.

5.4.2.4 Dependencies to other Components

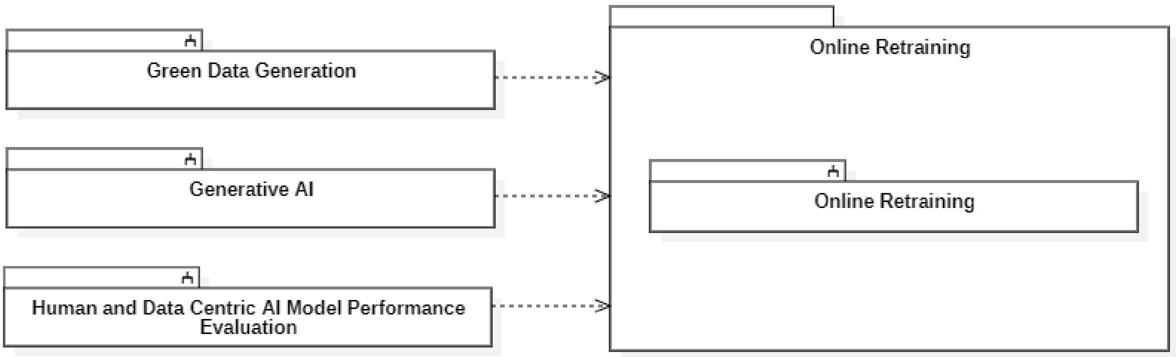

Figure 56 Online Retraining – Connectivity with other Components

Online Retraining node interacts with the following components:

- ***Green Data Generation (T3.1, T3.2) (From)***

The Online Retraining node receives data from the Green Data Generation, required to develop online re-training strategies.

- ***Generative AI (T3.6) (From)***

The Online Retraining node uses data from the Generative AI node, required to develop online re-training strategies.

- ***Human and Data Centric AI Model Performance Evaluation (T5.5) (From)***

The Online Retraining node uses indications on re-training needs from the human and data centric AI model performance node.

5.4.2.5 Addressed System Functional Requirements

SFR ID	Description
SFR22	Develop online re-training strategies for AI models

5.4.3 Federated Learning (FL) (T6.3)

5.4.3.1 Description

This node will implement a framework for federating training of models across heterogeneous data sources. The framework should promote efficient, secure and trustworthy model training across diverse data sources, as well as address potential data scarcity through transfer learning.

5.4.3.2 Main Functionalities:

- Federated learning across heterogeneous data sources
- Efficient, secure and trustworthy exchange of model parameters
- Adaptation of pre-trained models to new tasks / domains (transfer learning, meta-learning)
- Integrate interpretability methods adapted to federated learning

5.4.3.3 Internal Components

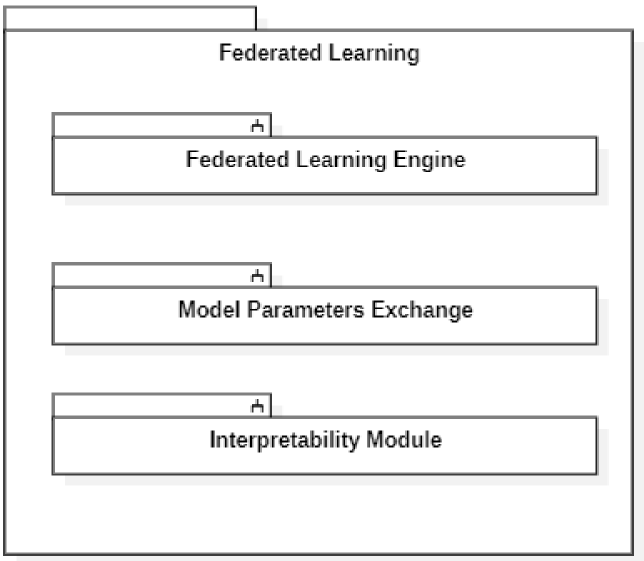

Figure 57 Federated Learning – Internal Components

- ***Federated Learning Engine***

FL engine, that enables massive FL applications to run simultaneously on edge networks. The key insight is to explore peer-to-peer (P2P) model to re-architect the centralized FL system design into a fully decentralized one.

- ***Model Parameters Exchange***

Model parameter exchange enables devices to share updates without sharing raw data, ensuring privacy. These parameters are aggregated to update a global model, which is redistributed for further training, fostering collaboration and scalability.

- ***Interpretability Node***

The interoperability module ensures seamless collaboration among devices, systems, or organizations with different architectures, frameworks, or protocols. It facilitates standardization for data formats, model architectures, communication protocols, and security measures, enabling diverse participants to work together.

5.4.3.4 Dependencies to other Components

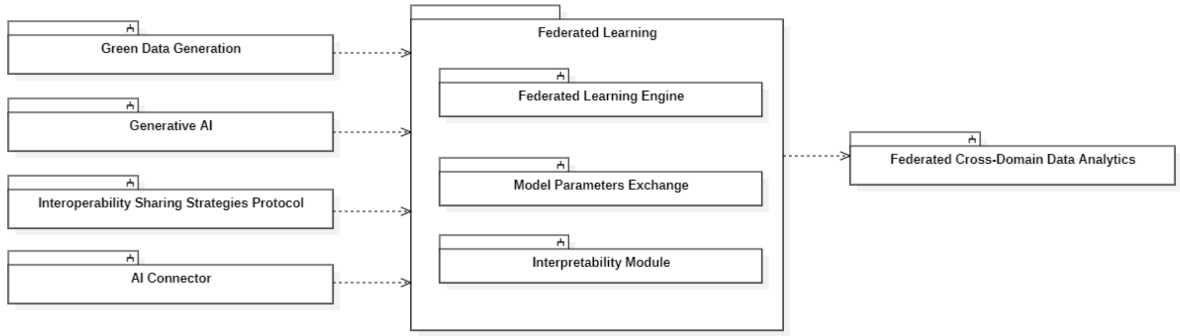

Figure 58 Federated Learning – Connection with other Components

- ***Green Data Generation (T3.1, T3.2) (From)***

It uses the use case-specific data sources, together with their data models and metadata.

- ***Generative AI (T3.6) (From)***

Federated Learning accesses training data possible from the Generative AI node to assist the overall approach.

- ***Interoperability Sharing Strategies/ Protocol (T5.2) (From)***

Federated Learning will use the Data Sharing strategies for interoperability, since it relies on data source from multiple data spaces and data lakes.

- ***AI Connector (T5.6) (From)***

AI-boosted connectors will be used for data transformation, secure and privacy-preserving data sharing.

- ***Federated Cross-Domain Data Analytics (T6.7) (To)***

Federated Learning will provide techniques to the Federated cross-domain data Analytics node to extract analytics to the developed models.

5.4.3.5 Addressed System Functional Requirements

SFR ID	Description
SFR20	Enable federated learning of AI models from heterogeneous data sources
SFR39	Create federated Digital Twin models

5.4.4 Data Aggregation, Analytics, and Visualization (DAAV) (T6.4)

5.4.4.1 Description

The DAAV node is structured to comprehensively address the requirements of data (D) aggregation (A), analytics (A), and visualization (V) across diverse domains. By taking advantage of state-of-the-art technologies and best practices, each component within this node will facilitate data management and analytics leading to visual representations of the result of this process.

5.4.4.2 Main Functionalities

This node will handle the output provided by WP3 by developing an ordered data flow which includes:

- ***Aggregation:*** Allows the connection to multiple sources of data to retrieve the information out of it to later transform it if it is necessary.
- ***Analytics:*** Involves extracting insights out of the data obtained from the previous Aggregation stage.
- ***Visualization***: Allows to inspect the analytics composed during the second stage of the node (Data Analytics).

5.4.4.3 Internal Components

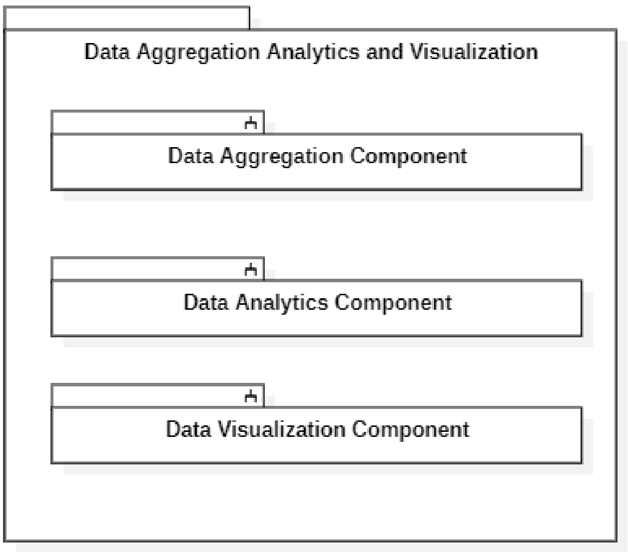

Figure 59 Data Aggregation, Analytics, and Visualization (DAAV) – Internal Components

- ***Data Aggregation component***

Primary source of data for the DAAV node, it involves both the connection and the retrieval of the information to later transform it, if necessary.

- ***Data Analytics component***

Intermediary step of the DAAV involves extracting insights out of the data obtained from the previous Aggregation stage.

- ***Data Visualization component***

The final component of the DAAV node allows to inspect the analytics composed during the second stage of the node (Data Analytics).

5.4.4.4 Dependencies to other Components

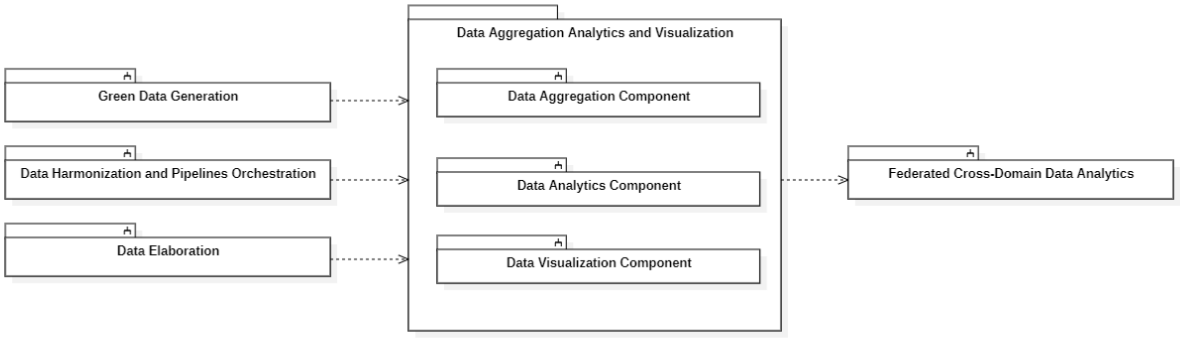

Figure 60 Data Aggregation, Analytics, and Visualization (DAAV) – Connectivity with other Components

- ***Green Data Generation (T3.1, T3.2) (From)***

The DAAV node uses data generated by the Green Data Generation node to perform aggregation, analytics extraction and visualization.

- ***Data Harmonization and Pipelines Orchestration (T3.3) (From)***

The DAAV node uses data products generated by the Data Harmonization and Pipelines Orchestration node.

- ***Data Elaboration (T3.5) (From)***

The DAAV node uses context-aware data generated by the Data Elaboration node.

- ***Federated Cross-Domain Data Analytics (T6.7) (To)***

The DAAV node provides analytics approaches as a basis to the Federated Cross-domain Data Analytics for the extraction of further analytics.

5.4.4.5 Addressed System Functional Requirements

SFR ID	Description
SFR42	Provide visual analytics and decision-making capabilities
SFR55	Extract decentralized analytics in real-time

5.4.5 Data Re-use System for Decision Support Systems (T6.5)

5.4.5.1 Description

Definition of novel data integration paradigms to be applied on energy and mobility data spaces to enable seamless data discovery and access across various data sources relevant to mobility and energy sectors by incentivizing data re-use for Decision Support Systems.

5.4.5.2 Main Functionalities

- Data retrieval from energy and mobility dataspace.
- Data manipulation for enabling a proper reuse.
- Possibility to apply AI, ML and analytics mechanisms to enable decision making.

Data Processes

The system will leverage the efforts from Data Creation (WP3) and Management (WP4) to retrieve heterogeneous data from other data spaces. It will implement human-centric methods that enable users to manipulate and format data effectively for use in mobility and energy data spaces. The formatted data will then be prepared for the application of Machine Learning (ML) and Artificial Intelligence (AI) mechanisms, facilitating decision-making processes. This module has to fetch data from the other data spaces. Communication protocols, APIs or centralized databases will have to be defined in order to allow the data exchange while ensuring security, traceability and accountability.

5.4.5.3 Internal Components

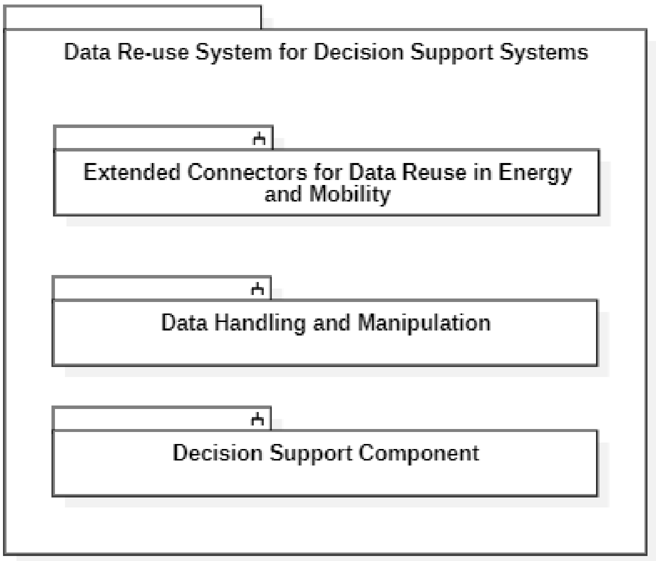

Figure 61 Data Re-use System for Decision Support Systems – Internal Components

- ***Extended Connectors for Data Reuse in Energy and Mobility***

Existing connectors will be further refined and extended in terms of data sharing methods in order to respect data format, metadata standards and semantic models and ensure data traceability and accountability for the mobility and energy sectors.

- ***Data Handling and Manipulation***

Component that receives in input the heterogeneous data from other data spaces and will apply algorithms to clean, sort, aggregate, infer and format data respecting data standards and accountability.

- ***Decision Support Component***

Component with integrated AI, ML and analytics methods in order to enable decision-making.

5.4.5.4 Dependencies to other Components

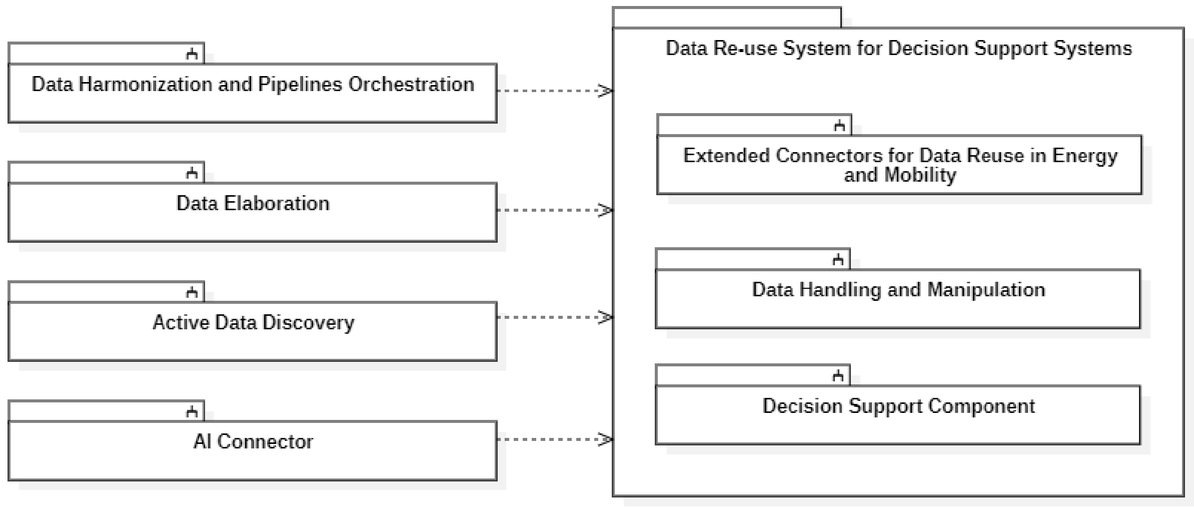

Figure 62 Data Re-use System for Decision Support Systems – Connection with other Components

- ***Data Harmonization and Pipelines Orchestration (T3.3) (From)***

The Data Handling and Manipulation internal component will rely on data products from the Data Harmonization and Pipelines Orchestration node.

- ***Data Elaboration (T3.5) (From)***

Extended connectors for data reuse in energy and mobility relies on development from data reuse capabilities and methods from Data Elaboration node.

- ***Active Data Discovery (T5.4) (From)***

Extended connectors for data reuse in energy and mobility will use data coming from the AI-based broker developed in the Active Data Discovery node.

- ***AI Connector (T5.6) (From)***

Extended connectors for data reuse in energy and mobility relies on development of the connectors for data retrieval developed in AI connector node.

5.4.5.5 Addressed System Functional Requirements

SFR ID	Description
SFR42	Provide visual analytics and decision-making capabilities
SFR55	Extract decentralized analytics in real-time
SFR56	Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing

5.4.6 AI-Enabled Data Quality Management (AI-DQM) (T6.6)

5.4.6.1 Description

The AI-Enabled Data Quality Management (AI-DQM) node is designed to evaluate and enhance the quality of data from diverse sources. Providing advanced artificial intelligence and machine learning techniques, this node focuses on assessing structured data correctness, consistency, and completeness. It implements automated processes to identify and rectify data issues, ensuring that the data is prepared for effective analysis and decision-making.

5.4.6.2 Main Functionalities

The AI-DQM node will provide a diverse set of features, all of them related to the ecosystem of ML pipelines:

- ***Data Quality Evaluation:*** Assess the quality of data across multiple dimensions using AI/ML techniques.
- ***Data Cleaning and Transformation:*** Automatically detect and correct data quality issues, including missing values, duplicates, and inconsistencies.
- ***Quality Improvement Strategies:*** Implement AI-driven strategies to enhance data quality based on continuous evaluation.

Data processes

The flow of data inside the AI-DQM node will potentially be divided into a sorted list of steps focusing on an Initial Quality Assessment of the raw data, as is provided, and a Final Quality Assessment after the main functionalities of the AI-DQM are applied:

- Data Ingestion
- Initial Quality assessment
- Definition of Data Quality Metrics relevant to the context
- Parametrisation of Data Quality Metrics by means of Data Quality Rules
- Cleaning & Quality Improvement
- Final Quality assessment

5.4.6.3 Internal Components

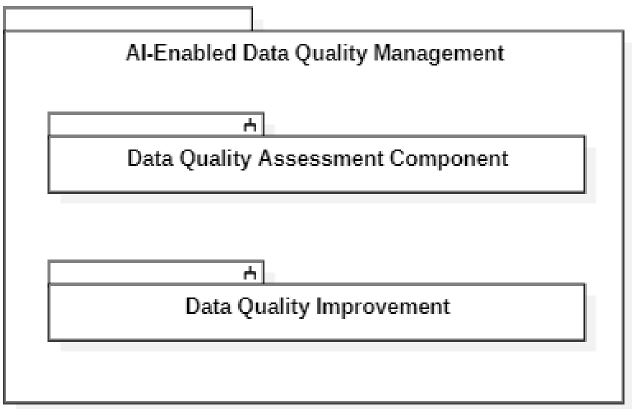

Figure 63 AI-Enabled Data Quality Management (AI-DQM) – Internal Components

- ***Data Quality Assessment Component***

This component evaluates the quality of incoming data based on predefined standards depending on each Use Case. In terms of structured datasets these procedures can use AI/ML techniques to detect anomalies and asses the overall quality of the data. This component will be used twice. First on receiving the raw data input and second after applying the Data Quality Enhancement techniques.

- ***Data Quality Improvement***

The Data Quality Improvement component automates the process of cleaning, transforming, and enhancing data quality using a combination of traditional data cleaning methods and advanced machine learning (ML) techniques. This component addresses issues such as missing values, duplicates, inconsistencies, and outliers, ensuring that data is accurate and reliable for analysis and decision-making.

5.4.6.4 Dependencies to other Components

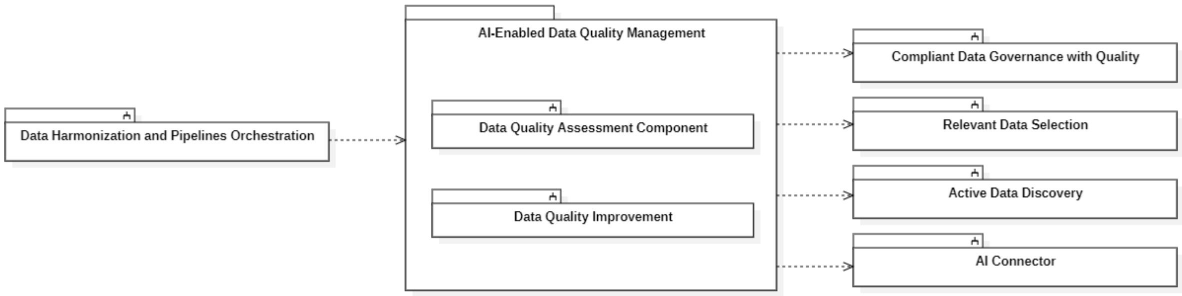

Figure 64 AI-Enabled Data Quality Management (AI-DQM) – Connection with other Components

- ***Data Harmonization and Pipelines Orchestration (T3.3) (From)***

The AI-DQM node receives data products from the Data Harmonization and Pipelines Orchestration node, which will be evaluated according to their quality.

- ***Compliant Data Governance with Quality (CompliantDQ) (T3.7) (To)***

The AI-DQM node will provide the quality assessment outcomes to the CompliantDQ node, to be used by the data governance methods that require the quality assessment.

- ***Relevant Data Selection (T4.6) (To)***

The AI-DQM node provides the quality assessment results to the Relevant Data Selection node, so that can use this information in the similarity and data disposal components.

- ***Active Data Discovery (T5.4) (To)***

The AI-DQM node provides the quality metrics to the Active Data Discovery node to take into account the quality during the extraction of recommendations.

- ***AI Connector (T5.6) (To)***

The AI-DQM node will provide the AI Connector node with automated data quality analysis methods, to be used during the data transactions.

5.4.6.5 Addressed System Functional Requirements

SFR ID	Description
SFR6	Prioritize data based on their quality metrics

SFR53	Evaluate continuously the quality of data
SFR54	Implement strategies to improve data quality

5.4.7 Federated Cross-Domain Data Analytics (T6.7)

5.4.7.1 Description

The Federated cross-domain data analytics node aims to achieve energy efficiency in diverse industrial settings by enabling analytics across multiple data domains without the need for centralized data storage, in alignment with the Green Deal initiatives and sustainability goals. The component will involve ML models as part of the PLIADES framework, able to optimize critical assets monitoring and subsequently reduce energy expenditure. The federated approach, testing, and evaluation will ensure the applicability of the component across multiple domains.

5.4.7.2 Main Functionalities

The main functionalities of the component are:

- Perform analytics on datasets from multiple domains without centralizing data
- Real-time insights and analytics to assist business decisions
- Trained ML models applicable in multiple domains

Data Processes

The node will require a large amount of data input both from the WP3 and potentially from the Physical Layer. The services developed in WP3 will result in high-quality homogenized data (real or artificial) that will be used to train individual models of the node.

5.4.7.3 Internal Components

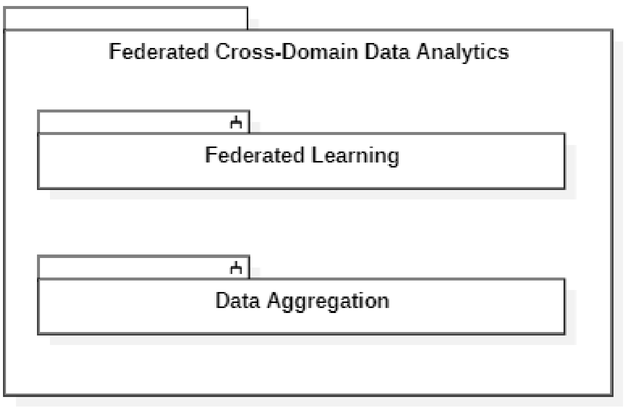

Figure 65 Federated Cross-Domain Data Analytics – Internal Components

- ***Federated Learning***

Training AI models across decentralized data sources.

- ***Data Aggregation***

Combining insights from multiple domains without moving data.

5.4.7.4 Dependencies on other Components

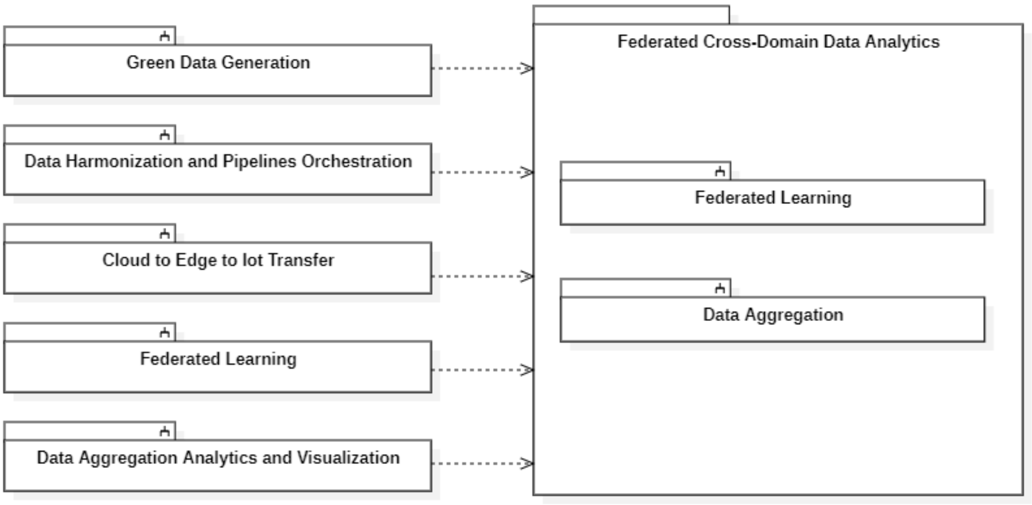

Figure 66 Federated Cross-Domain Data Analytics – Connection with other Components

- **Green Data Generation (T3.1, T3.2) (From)**

The Federated Cross-domain Data Analytics node is expected to receive as input processed real data and raw data from the physical world originating from the Green Data Generation node.

- ***Data Harmonization and Pipelines Orchestration (T3.3) (From)***

The Federated Cross-domain Data Analytics node will receive as input data products that have been processed, filtered and contains insightful metadata to extract analytics from multiple domains.

- ***Cloud to Edge to Iot Transfer (T4.2) (From)***

The Federated Cross-domain Data Analytics node will utilize decentralized data transfers from the Cloud to Edge to IoT transfer node, that will be used for federated data analytics extraction.

- ***Federated Learning (FL) (T6.3) (From)***

The Federated Cross-domain Data Analytics node will receive the Federated Learning techniques and will extract analytics on these models that are developed in the FL module.

- ***Data Aggregation, Analytics, and Visualization (DAAV) (T6.4) (From)***

The Federated Cross-domain Data Analytics node will use the analytics approaches developed in DAAV node, as a basis to extract energy-efficient cross-domain analytics.

5.4.7.5 Addressed System Functional Requirements

SFR ID	Description
SFR20	Enable federated learning of AI models from heterogeneous data sources
SFR42	Provide visual analytics and decision-making capabilities
SFR55	Extract decentralized analytics in real-time

5.5 Functional Nodes that Interact Directly with End-Users

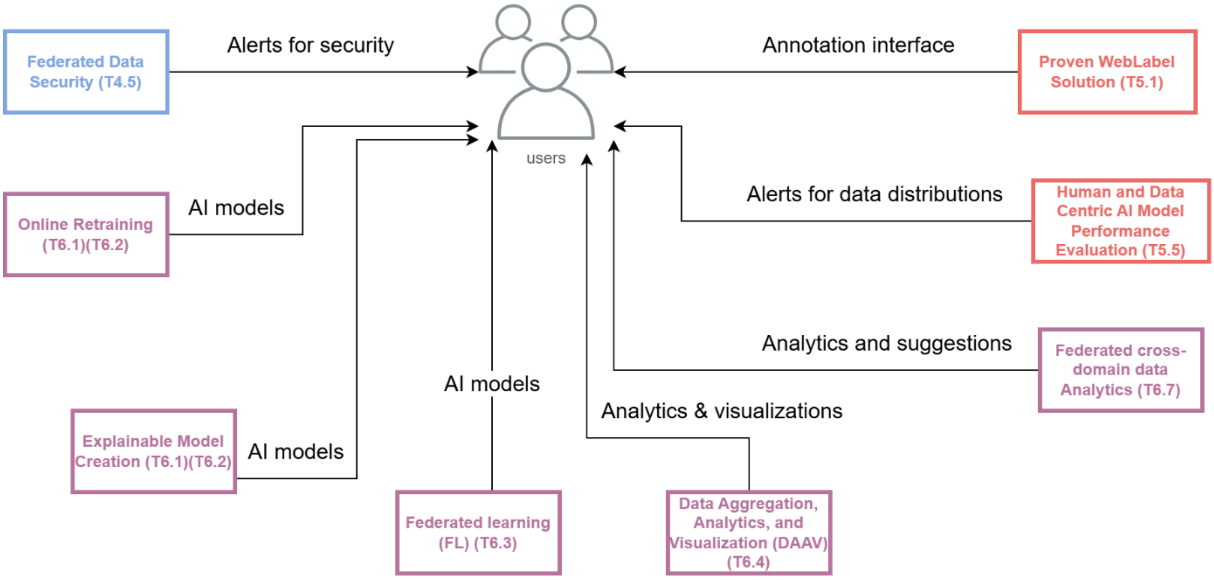

Figure 67 Functional nodes that interact directly with the end-user

- ***Federated Data Security (T4.5)***

Provides alert regarding suspicious activity and potential security threats using ML algorithms.

- ***Proven WebLabel Solution (T5.1)***

Direct interfacing with the end-user for the data annotation and visualization of annotations.

- ***Human and Data Centric AI Model Performance Evaluation (T5.5)***

Provides alerts to the end-user when data distributions are too different. Also, provides the end-user with an interface to perform actions, such as selecting relevant incoming datasets to improve the model (due to poor model performance or other reasons), correcting model predictions for specific samples or providing other useful feedback for model retraining.

- ***Explainable Model Creation (T6.1, T6.2)***

Extract new AI models that also provide explanations to the end-user in textual and/or visual format.

- ***Online Retraining (T6.1, T6.2)***

Extracts new AI models based on re-training strategies that have been generated.

- ***Federated Learning (FL) (T6.3)***

Federated Learning will share updated models for another Use Case participants to apply it to their data, by means of transfer learning

- ***Data Aggregation, Analytics, and Visualization (DAAV) (T6.4)***

The DAAV node allows to visualize and inspect the analytics composed.

- ***Federated Cross-Domain Data Analytics (T6.7)***

The output of this node will be trained models in a federated way, so that multiple cross-domain applications can be supported (e.g., identical machine equipment used in different industrial sectors). Also, output will be accurate suggestions about equipment replacement, recycling, repair, etc., to avoid costly unexpected failures resulting in larger energy consumption.

6 Technical Specifications

This chapter introduces the technical specifications of the PLIADES platform. Technical specifications can be divided into three categories: hardware, software and performance requirements for each functional component (PLIADES node). The hardware specifications of the components for carrying out the project's tasks are referred to as hardware requirements. The requirements of the platform's software components are referred to as software requirements. Performance requirements are the platform's quantitative capabilities that set the marks for successful/acceptable functionalities.

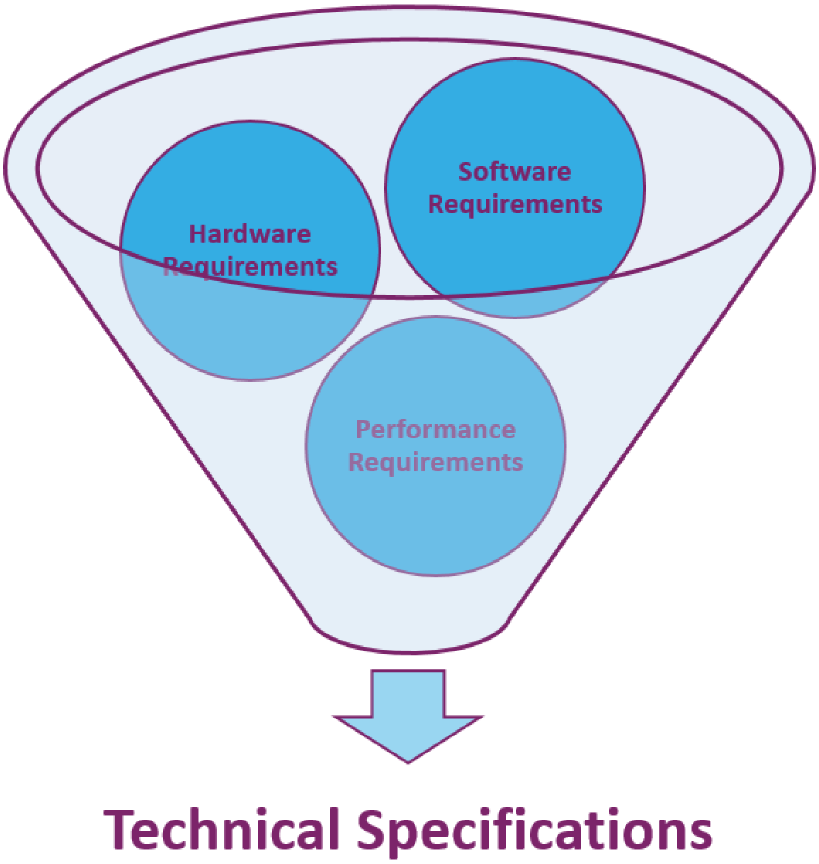

Figure 68 Illustrative representation on how the technical specifications are broken down

In summary, in Section 6.1 we present the mapping procedure from User Requirements to technical specifications. In Section 6.1 there are three sections included (Sections 6.1.1, 6.1.2, 6.1.3) which outline the performance, hardware and software requirements in detail. The main topic of Section 6.2 is the prioritization of the functional requirements, which are derived from the user requirements' priority.

6.1 Mapping of the User Requirements to Technical Specifications

There are two steps involved in mapping user requirements to technical specifications. The user requirements are first broken down into distinct use cases and their corresponding sub-use cases. Every sub-use case is examined in terms of a set of functional requirements, each of which is closely linked to particular performance requirements. The latter is the interface link that converts technical specifications into user requirements. The performance requirements act as a bridge that connects the technical specifications to the user needs. They ensure that the technical aspects, like hardware capabilities, software nodes, and system architecture are aligned with what the users expect in terms of performance. This breakdown helps in clearly identifying how the system is expected to behave in various situations, ensuring that all user needs are captured comprehensively.

Since the mapping process has been stabilized by combining the functional and performance requirements, all of the sub-use cases have been examined in terms of requirements in the second stage. Below is an example of the first mapping step for SubUC4.1, which is summed up in Table 4. This illustration relates to the sub-use case and user requirements for the AI-based ADAS development. The rest of the sub-use cases are described in Section 13, Annex III: Mapping of the User Requirements to Technical Specifications.

The process of mapping the software nodes and the intricacy of their connections is depicted in Figure 69. Our platform is clearly complex, because every user requirement is broken down into sub-use cases, each of which is connected to a number of technical specifications and functional requirements, and vice versa.

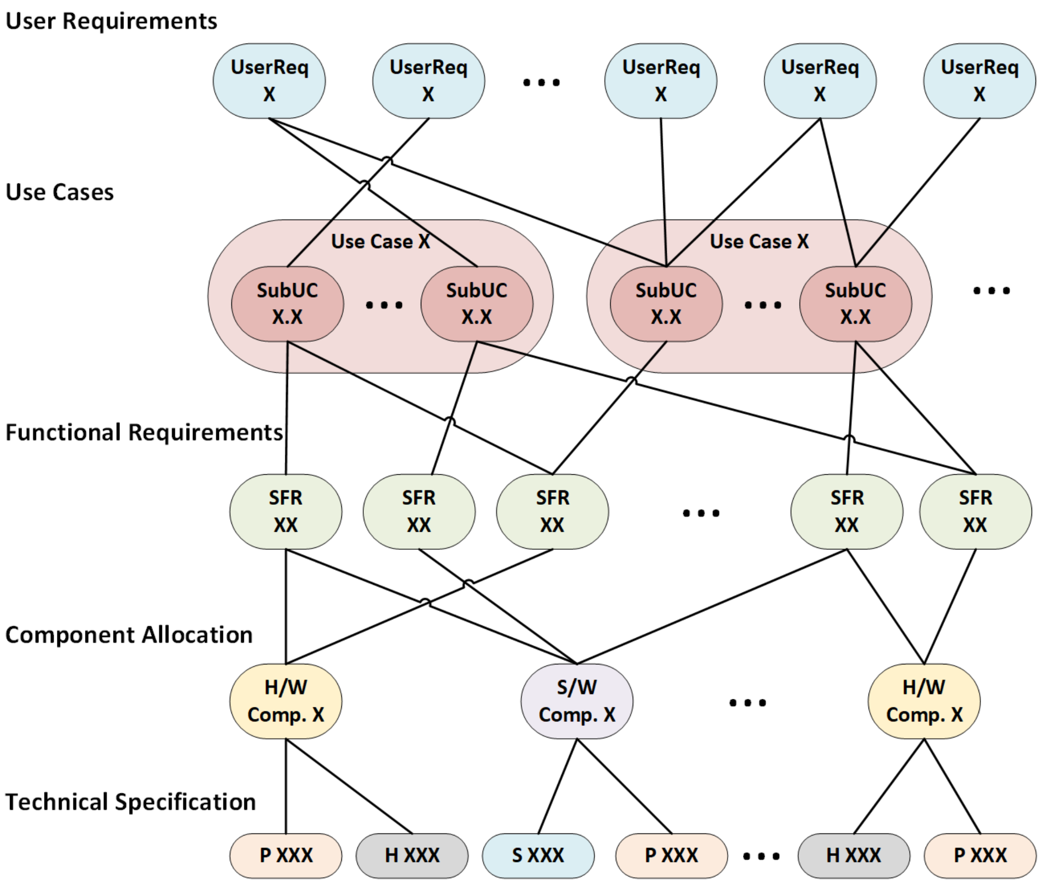

Figure 69 The mapping procedure and the complexity of the interconnections among the nodes

Table 4 Summary of Mapping of User Requirements to Technical Specifications for SubUC4.1

	High Priority	Medium Priority	Low Priority
Technical specifications for SubUC4.1: AI-based ADAS development			
Related Requirements -Functional: *UR16*, *UR18*, *UR25*, *UR27*, *UR34*, *UR35*, *UR36*, *UR37*, *UR38*, *UR40*, *UR41*, *UR42*, *UR43*, *UR44*, *UR45*, *UR46*, *UR47*, *UR48*, *UR49*, *UR50*, *UR51*, *UR52*, *UR53* **Non-Functional:** *UR81*, *UR82*, *UR83*, *UR84*, *UR85*, *UR86*, *UR87*, *UR88*, *UR89*, *UR90*			

System Functional Requirements	Performance requirements	Hardware requirements	Software requirements
SFR1 - Collect raw data from multiple sources	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, sensor APIs, hardware drivers
SFR3 - Synchronize data based on timestamps	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, access to NTP server
SFR5 - Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR6 - Prioritize data based on their quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR7 - Apply harmonization pre-processing techniques to ensure data consistency	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR9 - Generate annotation labels for data in a semi-automated manner	>90% confidence	Increased GPU capabilities	Python, Pytorch
SFR11 - Allow human validation of dataset labels with low confidence	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser

SFR12 - Inject human knowledge into data annotation and metadata creation processes	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR13 - Annotate multimodal data	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR14 - Visualize existing annotations of multimodal data	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR15 - Convert data between different formats	Real-time	Basic CPU/GPU capabilities	Python, REST API
SFR18 - Interface to collect human feedback for improving AI performance	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR19 - Create artificial data from raw data	Similarity to real data > 90%	High GPU capabilities	Python, Pytorch, Simulation engines
SFR20 - Enable federated learning of AI models from heterogeneous data sources	TBD according to model types	High GPU capabilities	Python, REST API
SFR21 - Provide explanations to AI models' inference	TBD according to model types	High GPU capabilities	Python, Pytorch
SFR22 - Develop online re-training strategies for AI models	TBD according to model types	High GPU capabilities	Python, Pytorch
SFR24 - Analyze and track data transactions	Real-time	Basic CPU/GPU capabilities	Python, Go

SFR25 - Anonymize or pseudonymize data by removing personally identifiable information	Accuracy >95%	Basic CPU/GPU capabilities	Python, Go
SFR26 - Ensure governance and rules in the control plane	Full assurance	Basic CPU/GPU capabilities	Python
SFR27 - Enforce compliance of data transactions according to legal and ownership restrictions	Full compliance	Basic CPU/GPU capabilities	Python, Go
SFR28 - Store sensitive data with restrictions	N/A	Basic CPU/GPU capabilities	Python, Go
SFR29 - Provide decentralized data connectors for secure, private, and sovereign data transfer between cloud and edge	N/A	Basic CPU/GPU capabilities	Python, Go
SFR31 - Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors	Near real-time	Basic CPU/GPU capabilities	Python, Go
SFR32 - Provide traceability and transparency for transactions in the entire network	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR33 - Identify security vulnerabilities in data space architecture and protocols	Accuracy > 95%	Basic CPU/GPU capabilities	Python, Go, C++
SFR35 - Analyze and prioritize threats within a dataspace	Real-time	Basic CPU/GPU capabilities	Python
SFR36 - Identify personal and non-personal data	Accuracy > 90%	Basic CPU/GPU capabilities	Python
SFR37 - Generate data flow maps with the entire lineage of each dataset	N/A	Basic CPU/GPU capabilities	Python, C++
SFR39 - Create federated Digital Twin models	N/A	High GPU capabilities	Python, Pytorch, Simulation engines
SFR40 - Enable cross-domain data discovery based on context and predefined criteria	Accuracy of suggestions > 90%	High GPU capabilities	C#, .NET, Python

SFR42 - Provide visual analytics and decision-making capabilities	Near real-time	Basic CPU/GPU capabilities, GUI-capable device	Python
SFR43 - Extract similarity metrics for multimodal data	Near real-time	High GPU capabilities	Python
SFR44 - Discard data based on similarity metrics	Real-time	Basic CPU/GPU capabilities	Python
SFR46 - Allow data exchange with ontology-based components	N/A	Basic CPU/GPU capabilities	Python, Go
SFR47 - Perform cross-domain adaptation between datasets	Depends on the dataset	High GPU capabilities	Python
SFR49 - Allow queries to explore existing databases and internal relations	Query processing time < 5s	Basic CPU/GPU capabilities	Python, Go
SFR50 - Provide strategies for data interoperability between data spaces	N/A	Basic CPU/GPU capabilities	Python
SFR51 - Receive natural language inputs to perform queries in Graphical Databases	Process query time < 1s	High GPU capabilities	Python
SFR52 - Provide optimal recommendations between datasets and data consumer needs	Recommendation accuracy > 90%	High GPU capabilities	C#, .NET, Python
SFR53 - Evaluate continuously the quality of data	Real-time	High GPU capabilities	Python
SFR54 - Implement strategies to improve data quality	N/A	Basic CPU/GPU capabilities	Python
SFR55 - Extract decentralized analytics in real-time	Real-time	High GPU capabilities	Python, C++
SFR56 - Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing	N/A	High GPU capabilities	C#, .NET, Python

6.1.1 Performance Requirements

The performance requirements are essential for developing and designing the PLIADES platform, ensuring that it meets its intended purpose. The corresponding functional requirements are defined both quantitatively and qualitatively. Specifically, the performance requirements serve as a bridge for determining the hardware and software requirements and specify the capabilities that the PLIADES platform should maintain. They include observations about the platform's operational limitations, technical and physical capabilities, and time response. This also includes the platform's responsiveness, ensuring it performs efficiently under all conditions. A list of precise performance requirements focused on the fundamental hardware and software elements of PLIADES platform has been established, in conjunction with all functional requirements.

6.1.2 Hardware Requirements

Following the identification of the primary hardware components, the precise specifications for each will be developed. The physical architecture of the PLIADES platform is followed in the organization of these requirements, which relate to the hardware components that will be used to implement and validate the PLIADES platform. The required processing units and their capabilities are identified from the needs extracted by the corresponding software requirements, and specifically the nodes which requires some kind of infrastructure to execute and integrate with the overall PLIADES platform.

The final version of the hardware requirements will be included in the list of hardware requirements that will be contained in the future integration deliverable. An initial abstract outline of the hardware needs will lead to a more complete detailed catalogue, where these components will be evaluated for their performance and their cost-effectiveness. This process will help to identify any operational constraints or potential areas for optimization, such as power consumption, processing power and data throughput.

6.1.3 Software Requirements

The software requirements are essential for determining the needs that the PLIADES platform must address. They are made of the identified essential functional elements that must be developed in order to transform the stand-alone data operations into fully operational data pipelines. The software will act as the driving force for the communication and coordination of various data operations. Along with serving particular routines that satisfy the demands of the sub-use cases, they are also accountable for meeting the performance requirements in terms of execution time and accuracy. Each sub-use case presents unique operational demands, so the software requirements must cater to these specific needs by defining how the software will handle and execute these tasks effectively. The final version of the software requirements will be included in the integration deliverable, which is currently being constructed.

6.2 Prioritization of the System Functional Requirements

The aim of the analysis presented here is to prioritize the functional requirements to be implemented in accordance with the User Requirements' established priorities. Entities with 'High', 'Medium', or 'Low' priority, according to particular rules make up the prioritization schema used in this process. In order to accomplish a thorough prioritization, the following standards were taken into account:

- According to the D2.2 "User Requirements and Use Cases", the priority ranking of the identified use cases, sub-use cases, and user requirements.
- The frequency with which each functional requirements appears in the particular sub-use cases, as shown in the tables above. Given that it appears to be a functional requirement

for handling numerous sub-use cases, it should be given top priority because it shown great significance and relevance to the project.

- The dependency on critical (high priority) functional requirements required to complete a specific sub-use case.

Following the aforementioned procedure, Section 12 contains a summary table that shows the priority of each system functional requirement. By assessing the resulting table, it is realized that there are 56 identified system functional requirements, of which 37 have high priority, 16 have medium priority, and 3 remain low priority. Each priority's overall percentage coverage is displayed in the diagram below. It is ensured that software and hardware requirements necessary for the fulfillment of the high priority sub-use cases will be developed early on by prioritizing the functional requirements, which are closely linked to the technical specifications.

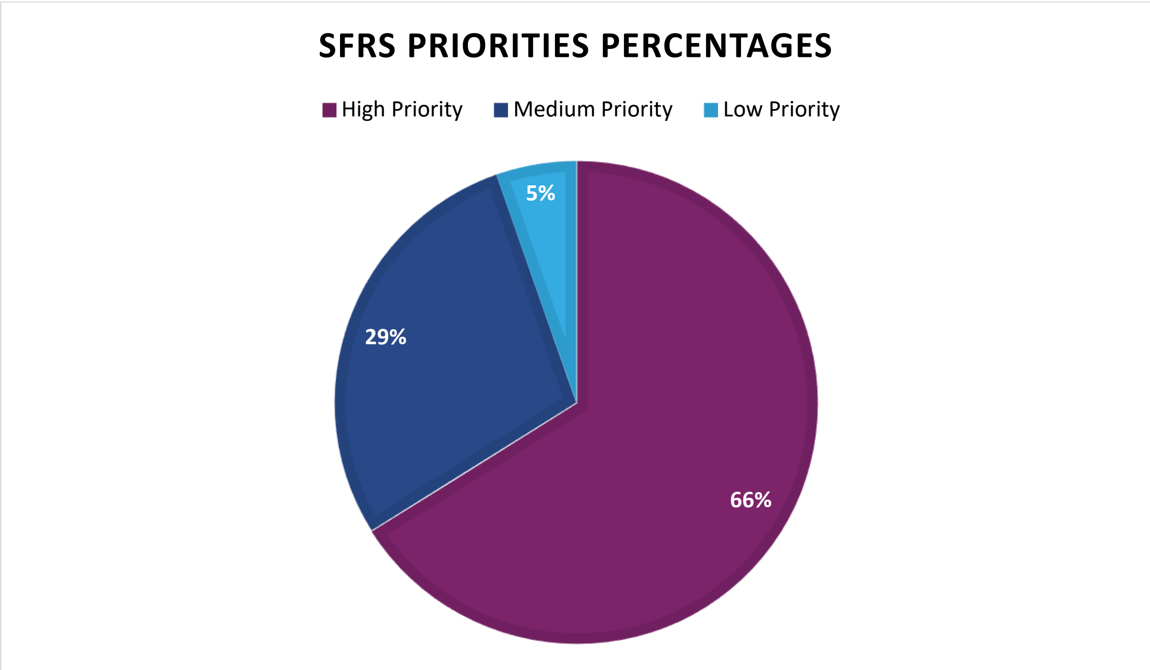

Figure 70 System Functional Requirements Priorities Percentages.

7 Integration with the IDSA architecture

The integration of the PLIADES framework with the International Data Spaces Association (IDSA) architecture is essential to achieve seamless, secure, and standardized data exchange across diverse domains. Leveraging IDSA's principles and technical components, the PLIADES system ensures data sovereignty, interoperability, and compliance with European data protection standards. This section details the mechanisms, roles, and architecture necessary for embedding IDSA connectors and facilitating data exchange within a data space and between different data spaces.

The IDSA architecture comprises of core components, such as the IDS Connectors, the Metadata Broker and the Clearing House, which will serve as the foundation to build the PLIADES architecture and ensuring compatibility with the existing framework (and upcoming updates). The basic interaction of technical components in the system layer, based on the current IDS architecture (IDS RAM 4.0) is depicted below, in Figure 71.

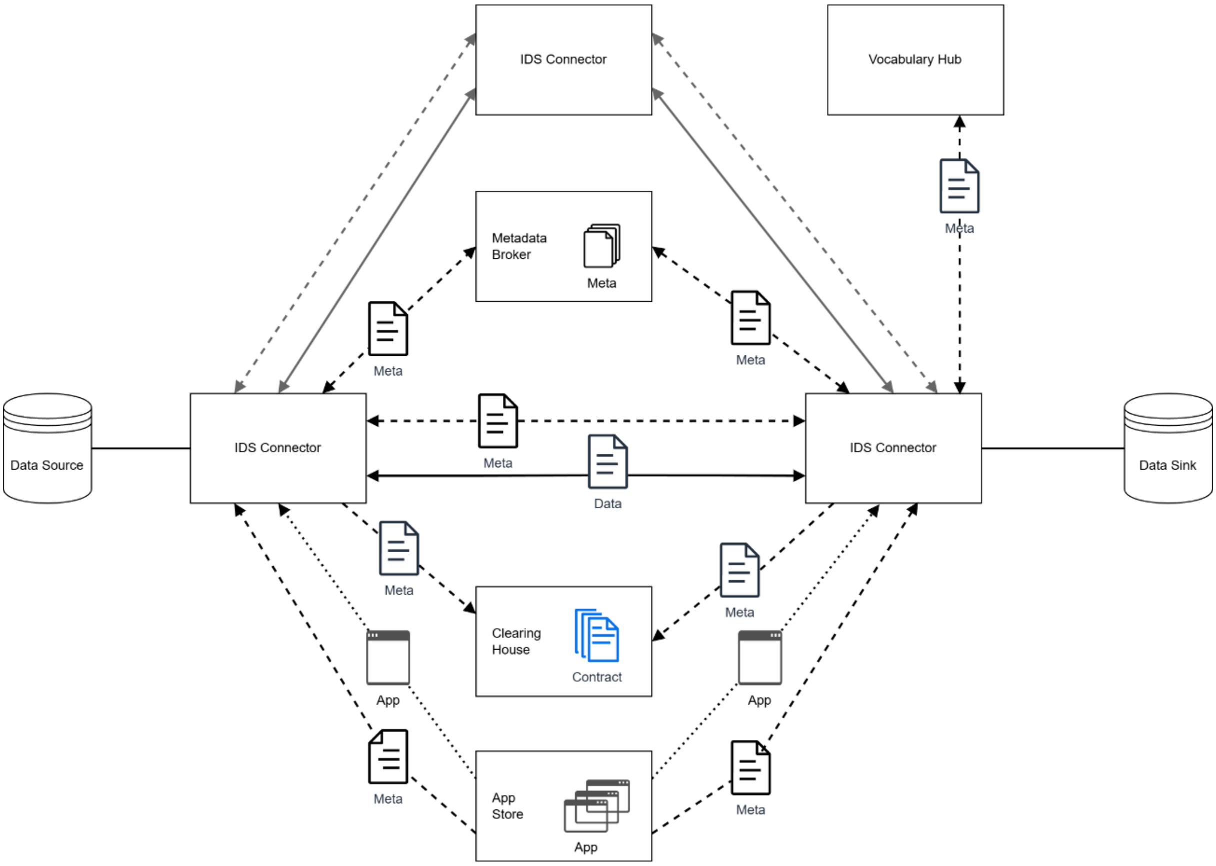

Figure 71 Interaction of IDSA technical components

One the one hand, PLIADES will address data sharing within a single dataspace. In this case, the architecture will remain consistent and the software components developed through PLIADES will either replace or extend the existing capabilities. As shown in Figure 72, PLIADES connectors will communicate with existing IDS connectors, allowing the intra-dataspace data exchange. In addition, existing metadata brokers will be replaced/enhanced by the AI-based PLIADES brokers, which will allow improved discoverability of data, using the corresponding metadata.

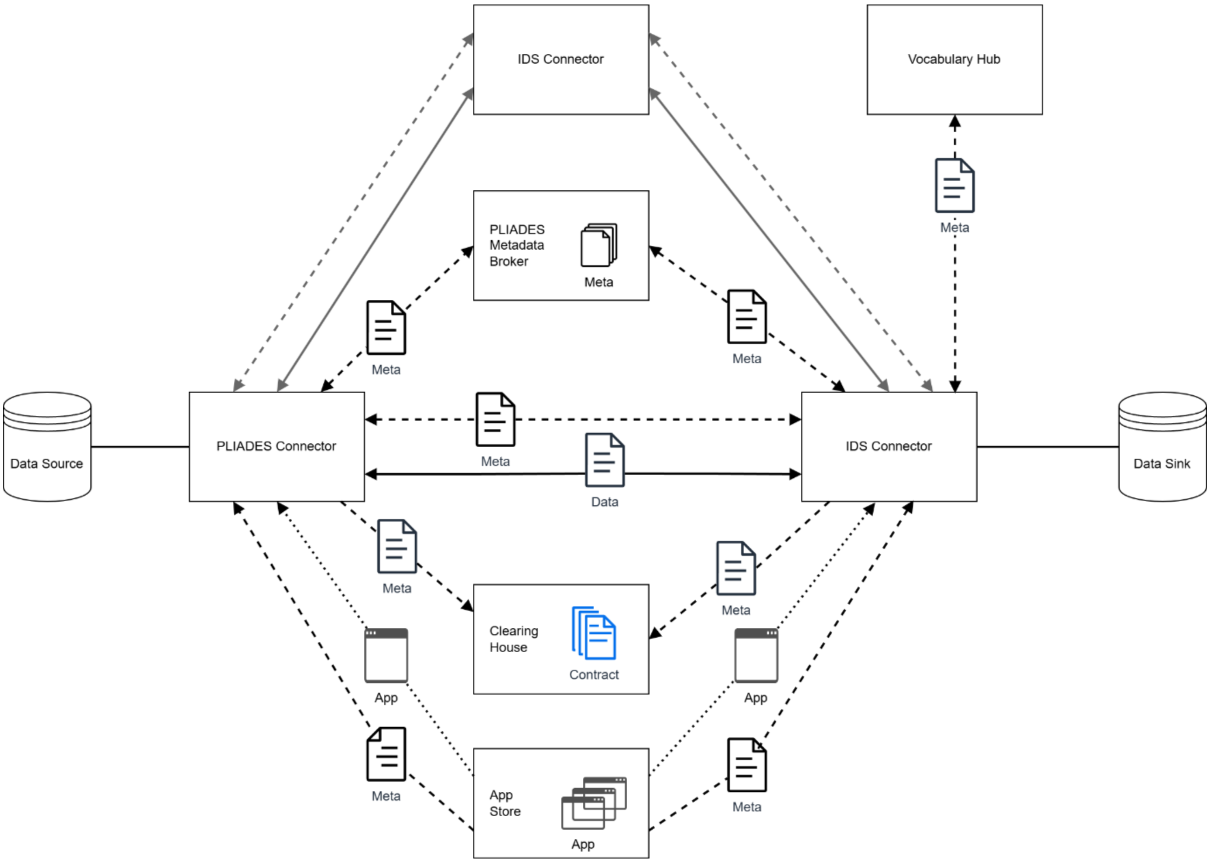

Figure 72 Intra-data space PLIADES architecture

On the other hand, PLIADES will address data sharing between different dataspaces. This will be implemented, following the architecture depicted in Figure 73. Without affecting the functionality of existing IDS connectors, for intra-data space operations, the PLIADES connectors of data space A will be able to communicate the PLIADES connectors of data space B. Also, the cross-domain PLIADES broker will provide suggestions regarding the available data on data space A and B simultaneously.

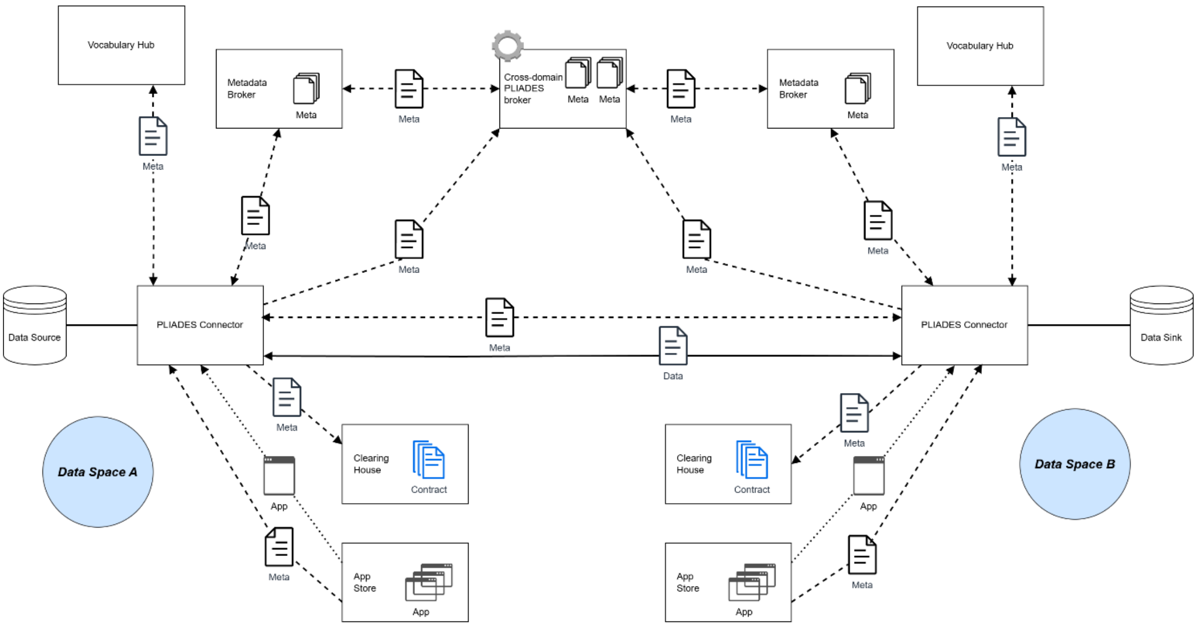

Figure 73 Cross-data space PLIADES architecture

Below, we provide the connection between the PLIADES software nodes and existing components of the IDS architecture for specific scenarios that will be enriched and improved with the provided functionalities.

7.1.1 Data generation by the data provider

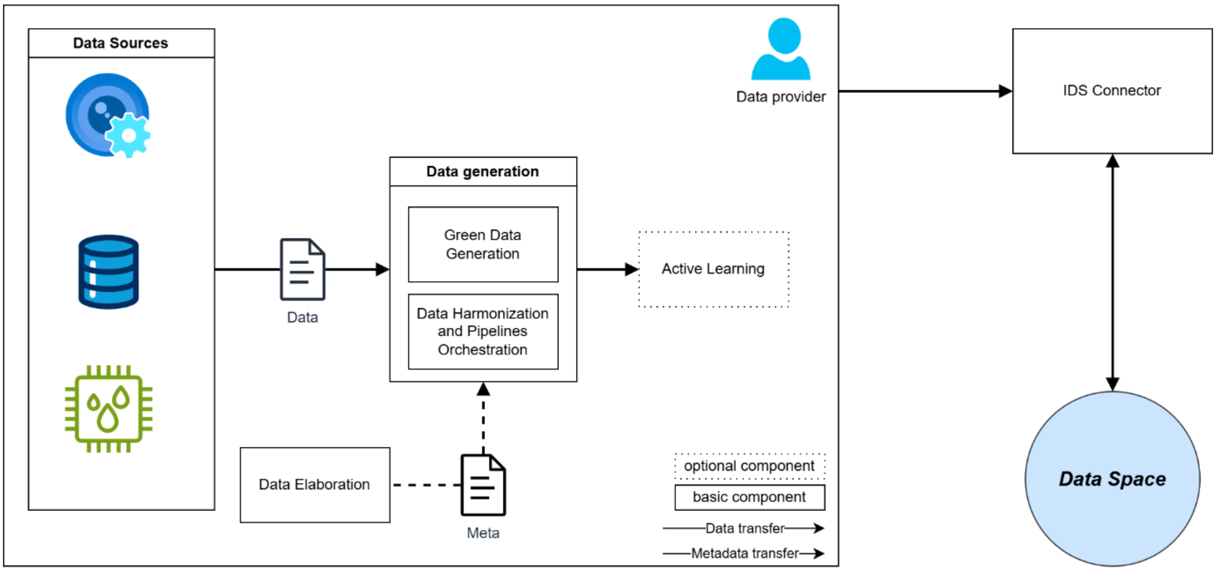

Figure 74 Data generation by the data provider

As shown in Figure 74, the data provider, who is also the data creator, executes specific pipelines before providing their data to a data space through the designed connector. These steps include:

- the filtering and basic pre-processing steps, which will reduce the amount of data without compromising their quality of information
- the integration of extended metadata, which will assist further data re-use in the dataspace and will lead to enhanced discoverability
- optional semi-automated annotation methods through active learning techniques, to provide accurate labels for the generated data

7.1.2 PLIADES Applications

Various PLIADES software nodes will be used as data space applications, similarly to the existing IDS Apps. Existing IDS apps are distinguished in three types: i) Data app, ii) Adapter app, and iii) Control app, each performing a different task within the IDS ecosystem.

The existing architecture of IDSA App Store is depicted in Figure 75.

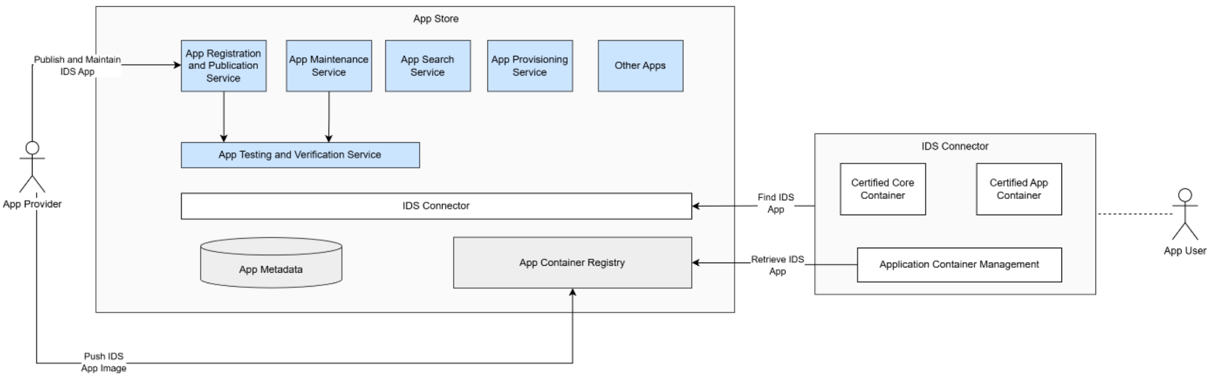

Figure 75 App Store architecture

Among the PLIADES software nodes described in Section 5, specific Data Apps have been identified that will extend the current capabilities of IDSA App Ecosystem and are displayed below in Figure 76 and annotated with green colour. PLIADES connectors will also interact with the IDS App Store and ensure compliance with existing practises.

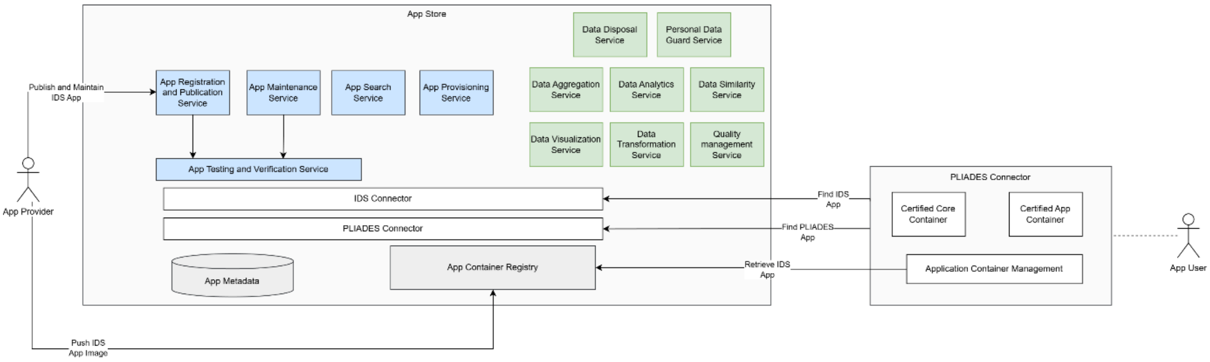

Figure 76 App Store with PLIADES Apps

7.1.3 PLIADES Connectors

The PLIADES platform includes specific connectors that will be developed and will be capable to replace existing IDS connectors, while providing further capabilities. According to the IDS RAM 4.0, the architecture of these connectors will use application container management technology to ensure an isolated and secure environment for individual IDS Apps and IDS Connector functionalities. The developed connectors will rely on the architecture of IDS connectors, as depicted in Figure 77.

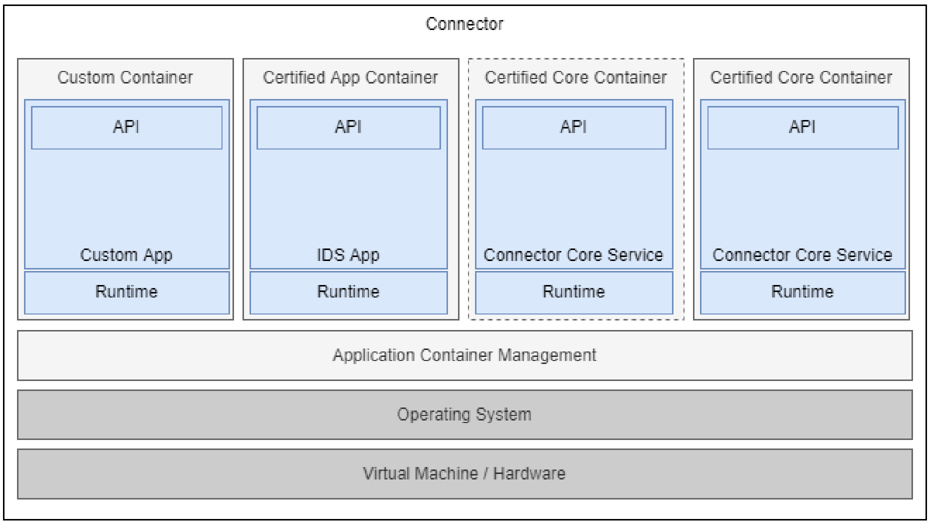

Figure 77 IDSA Connector architecture

Following the PLIADES software nodes identified in Section 5, below we list some indicative connectors and their functionalities:

- *Cloud to Edge to IoT Transfer*: Decentralized data connectors, which will allow data exchange between edge and cloud technologies.
- *AI Connector*: Data connectors for cross-domain data exchange, which will establish data transformation, security and privacy.
- *Data Re-use System for Decision Support Systems*: Extension of connectors to ensure traceability and accountability for the mobility and energy sectors.

8 Implementation

8.1 Introduction

This chapter comprises an overview of the PLIADES platform and its software nodes' requirements. Initially, we perform a brief description of the main technologies that most software nodes employ and introduce the source code management platform. The chapter continues with a description of the hardware requirements for the employed software tools, and continues with a detailed description of the main hardware infrastructure of the PLIADES platform, including a preliminary analysis for the computational resources needed. Finally, the integration of the PLIADES architecture with the existing IDSA architecture is described.

8.2 Development View

8.2.1 Node Development

Table 5 Summary of Node Development

Name	Owner	Programming Language/ Technology
Green Data Generation	CERTH, BOR	Python
Data Harmonization and Pipelines Orchestration	TECNALIA	Python
Active Learning	VICOM	Python
Data Elaboration	CERTH	Python
Generative AI	CERTH	Python
Compliance Data Governance with Quality (CompliantDQ)	EURECAT	Python/Go
Hyperledger Fabric Blockchain – Data Space Connector Interface (HFB Interface)	CEIT	Go
Cloud to Edge to IoT transfer	E@W	Python, Java, Go
Privacy-Enhanced Data Collaboration (PEDC)	EURECAT	Python, C++
Identity Access Management- Data Space Connector Decentralization IAM	CEIT	Go
Federated Data Security (FDS)	EURECAT	Python
Relevant Data Selection	TECNALIA	Python
Personal Data Guard	CERTH	Python

Proven Weblabel Solution	VICOM	Typescript/Angular
Proven Ontology Manager	VICOM	Python, GraphDB
Interoperability Sharing Strategies / Protocol	INNOVALIA	Python
AI-based Query System	VICOM	Python
Active Data Discovery	ATLANTIS	C#, .NET, Python
Human and Data Centric AI Model Performance Evaluation	TECNALIA	Python
AI Connector	ATLANTIS	C#, .NET
Explainable Model Creation	CERTH	Python
Online Retraining	CERTH	Python
Federated learning (FL)	EURECAT	Python
Data Aggregation, Analytics, and Visualization (DAAV)	EURECAT	Python
Data Re-use System for Decision Support Systems	E@W	Python
AI-Enabled Data Quality Management (AI-DQM)	EURECAT	Python
Federated cross-domain data Analytics	ATLANTIS	C#, .NET, Python

8.2.2 Source Code and Configuration Management

In software development the core of development management depends to a version control system which is essential in large scale projects, where multiple partners must contribute to the developing system. Such system is able to maintain current and older versions of files and folder structure, allowing multiple programmers to easily cooperate. Configuration management contains version control, and extends it by providing additional methods for project management, such as:

- Identification of Software Configuration Items (SCIs) and their versions through unique identifiers
- Control over performed changes through management or instance control
- Accounting of the state of individual components
- Auditing of selected versions (which are scheduled for a release) by a quality control team

For the development and management of the source code for all PLIADES nodes, Git will be used, which is a free and open-source distributed version control program.

8.3 Deployment View

8.3.1 Platform Hardware Architecture

The PLIADES hardware architecture is designed based on the analysis of the expected functionalities of the platform, including the system functional requirements, the available technological means, as well as the needs of each consortium member. The hardware components of PLIADES are represented by the following layers:

- Edge devices, including sensors and data gateways
- Data processing units
- Data storage and management units
- Networking infrastructure
- Visualization interfaces and devices

Below, we present the basic characteristics of the main hardware components of PLIADES. This infrastructure is provided by the consortium partners and can be employed as is, for the development and the validation of the PLIADES software nodes.

8.3.1.1 Edge devices

Edge devices act as the initial layer of interaction between the platform and the physical environment. These devices include IoT sensors and actuators that collect data from various sources, such as environmental, mobility, or industrial systems. Edge gateways are employed to perform initial data pre-processing, reducing the burden on downstream systems, and to ensure secure and efficient transmission of data to central processing units. These devices are often equipped with tamper-proof mechanisms and encryption capabilities to meet stringent security and compliance requirements, such as GDPR.

8.3.1.2 Data processing units

Data processing units are responsible for the computational needs of the platform, providing the resources necessary for AI model training, inference, and data analytics. They consist of high-performance computing nodes with GPUs or TPUs optimized for AI tasks, computer vision and/or natural language processing. This layer supports both onboard and cloud-based deployments, ensuring scalability and flexibility to meet varying workload demands.

8.3.1.3 Data storage and management units

The data storage and management units ensure secure, efficient, and scalable handling of diverse data types. These units include distributed databases for structured data and object storage for unstructured data. Localized storage nodes are employed in edge scenarios to minimize latency and support federated learning. The storage systems are designed to maintain high availability and fault tolerance, leveraging redundancy and automated backup mechanisms to safeguard against data loss.

8.3.1.4 Networking infrastructure

Networking infrastructure underpins the seamless connectivity between edge devices, data processing units, and storage systems. It leverages high-speed wired and wireless connections to facilitate low-latency communication. Secure data transfers are ensured through encryption protocols, virtual private networks (VPNs), and firewalls. The infrastructure is designed to be resilient, supporting dynamic routing and load balancing to handle varying data traffic loads and maintain uninterrupted operations.

8.3.1.5 Visualization interfaces and devices

Visualization interfaces and devices enable the interaction with the platform's outputs, making data and AI insights accessible to end-users. These devices mostly refer to the software nodes that were mentioned in Section 5.5 and have direct interaction with the end-user. The interfaces are designed for usability, incorporating responsive designs and customizable visualizations to cater to different user roles and preferences.

8.3.1.6 Hardware requirements per software node

Below, in Table 6 we present the minimum hardware requirements per software node. The hardware requirements per node analysis allow us to correctly estimate the necessary computational resources for the entire platform. In this document, we present the initial estimation for each software module, which is likely to be updated later on and reported in the WP7 “Deployment in Diverse Use Cases for Data Spaces Integration and Cross Domain Validation” deliverables. Online refers to the nodes that need to be performed in real-time, during data operations, such as data collection, data transmission, etc. On the other hand, offline refers to methods that can be performed ad-hoc in an offline manner.

Table 6 Hardware Requirements per Software Node

Software Node	Online/ Offline	Minimum hardware requirements
Green Data Generation	Online	NVIDIA GPU with 8GB memory, Multicore processor, 32GB RAM
Data Harmonization and Pipelines Orchestration	Online	Basic CPU/GPU processing
Active Learning	Offline	NVIDIA GPU with 8GB memory, Multicore processor, 32GB RAM
Data Elaboration	Online	NVIDIA GPU with 8GB memory, Multicore processor, 32GB RAM
Generative AI	Offline	NVIDIA GPU with 24GB memory, Multicore processor, 32GB RAM
Compliance Data Governance with Quality (CompliantDQ)	Online	Basic CPU/GPU processing
Hyperledger Fabric Blockchain – Data Space Connector Interface (HFB Interface)	Online	Basic CPU/GPU processing
Cloud to Edge to IoT transfer	Online	Basic CPU/GPU processing
Privacy-Enhanced Data Collaboration (PEDC)	Online	Basic CPU/GPU processing
Identity Access Management- Data Space Connector Decentralization IAM	Online	Basic CPU/GPU processing
Federated Data Security (FDS)	Online	Basic CPU/GPU processing

Relevant Data Selection	Online	Basic CPU/GPU processing
Personal Data Guard	Online	Basic CPU/GPU processing
Proven Weblabel Solution	Offline	Access to file and data server
Proven Ontology Manager	Online	GraphDB server
Interoperability Sharing Strategies / Protocol	Online	Basic CPU/GPU processing
AI-based Query System	Online	NVIDIA GPU with 12GB memory, Multicore processor, 32GB RAM
Active Data Discovery	Online	Basic CPU/GPU processing
Human and Data Centric AI Model Performance Evaluation	Online	Basic CPU/GPU processing
AI Connector	Online	Basic CPU/GPU processing
Explainable Model Creation	Offline	NVIDIA GPU with 8GB memory, Multicore processor, 32GB RAM
Online Retraining	Offline	NVIDIA GPU with 8GB memory, Multicore processor, 32GB RAM
Federated learning (FL)	Offline	Quad Core processor, 32GB RAM
Data Aggregation, Analytics, and Visualization (DAAV)	Offline	Basic CPU/GPU processing
Data Re-use System for Decision Support Systems	Offline	Basic CPU/GPU processing
AI-Enabled Data Quality Management (AI-DQM)	Online	Basic CPU/GPU processing
Federated cross-domain data Analytics	Offline	Basic CPU/GPU processing

8.3.2 Preliminary Analysis of Computational Resources

According to the requirements described in Section 8.3.1.6, most of the PLIADES software nodes require basic CPU/GPU processing capabilities, without any special needs. This implies that the platform can be used in any type of processing unit available, with recent and updated specifications. The nodes which have special requirements are the ones which are responsible for the training and the validation of AI models. Especially, LLM-based models or Generative AI models require a large processing capacity to operate efficiently and produce the desired outcomes. More specific details on the exact needs of each module will be define in later stages of the project and will be described through the WP7 deliverables, related to the integration of the nodes.

9 Conclusion

This report has presented the PLIADES conceptual architecture, offering a detailed overview of its modular structure, high-level functionalities, and interdependencies. Building on the user requirements and use cases outlined in D2.2 “User Requirements and Use Cases”, this deliverable has translated them into a hierarchical system architecture, forming the foundation for the implementation framework.

The PLIADES architecture follows a modular approach to optimize data lifecycles and enhance interoperability within data spaces. The analysis progressed from defining abstract modules to detailing system nodes and their roles in specific use case scenarios. A functional analysis has outlined how operations and processes are implemented through modular components, focusing on interactions, dependencies, and the mapping of system functional requirements to use cases.

Technical specifications – including performance, hardware, and software requirements – have been systematically defined and aligned with user needs. These specifications were organized into development and deployment views: the Development View outlines partner responsibilities and module development guidelines, while the Deployment View specifies hardware and software integration requirements.

In conclusion, the PLIADES framework architecture and specifications have been broken down into manageable, modular components to ensure scalability and adaptability. This deliverable represents the initial iteration of an evolving design process that will be refined through development and deployment feedback. Future updates and advancements will be documented in future deliverables in WP7 “Deployment in Diverse Use Cases for Data Spaces Integration and Cross Domain Validation”, driving the project’s mission of advancing AI-enabled data lifecycle management and interoperability.

10 References

[1] Rozanski, Nick, and Eóin Woods. Software systems architecture: working with stakeholders using viewpoints and perspectives. Addison-Wesley, 2011
[2] Richards, M., & Ford, N. Fundamentals of Software Architecture: An Engineering Approach. O'Reilly Media, 2021
[3] Rozanski, Nick, and Eóin Woods. Software systems architecture: working with stakeholders using viewpoints and perspectives. Addison-Wesley, 2011
[4] Richards, M., & Ford, N. Fundamentals of Software Architecture: An Engineering Approach. O'Reilly Media, 2021

11 Annex I: User Requirements

Table 7 Functional user requirements of the PLIADES platform

ID	Type	Description	Priority Level
UR1	Functional	Implement automated quality checks based on real-time machine data	HIGH
UR2	Functional	Enable flexible machine settings adjustments based on varying production requirements	MEDIUM
UR3	Functional	Provide adaptive control for material specifications based on data-driven insights	HIGH
UR4	Functional	Integrate quality data from previous production runs to improve current batch consistency	MEDIUM
UR5	Functional	Facilitate collaborative adjustments between operators and quality managers for improved outcomes	LOW
UR6	Functional	Use predictive algorithms to adjust machine parameters before defects occur	HIGH
UR7	Functional	Enhance data visualization tools for real-time monitoring of quality metrics	MEDIUM
UR8	Functional	Dynamically adjust production schedules to account for material and product specification changes	LOW
UR9	Functional	Integrate human-in-the-loop methodologies to refine AI algorithms	HIGH
UR10	Functional	Automate compliance checks for data access requests	MEDIUM
UR11	Functional	Provide users with real-time compliance notifications for data operations	HIGH
UR12	Functional	Implement pilot projects with early adopters in new markets	HIGH
UR13	Functional	Develop training programs to help clients utilize DENNDATA for new business opportunities	HIGH
UR14	Functional	Leverage AI to explore cross-industry data opportunities	LOW
UR15	Functional	Enable real-time collaboration tools for stakeholders working on new projects	LOW
UR16	Functional	Implement compression techniques to optimize data transmission	HIGH

UR17	Functional	Optimize edge computing for onboard filtering processes	MEDIUM
UR18	Functional	Automate identification and discard of noisy and low-quality multimodal data	HIGH
UR19	Functional	Develop semi-Automated methods for multimodal data quality estimation	MEDIUM
UR20	Functional	Automate metadata tagging during data collection	HIGH
UR21	Functional	Increase data discoverability through context-aware metadata	MEDIUM
UR22	Functional	Automate analysis for cultural issues of human tracking data	LOW
UR23	Functional	Automate annotation processes for multimodal data	HIGH
UR24	Functional	Implement compression and filtering techniques to reduce data storage needs	MEDIUM
UR25	Functional	Automate identification and discard of redundant data	HIGH
UR26	Functional	Implement advanced encryption protocols for all data	HIGH
UR27	Functional	Automate monitoring of data compliance with privacy and quality standards	MEDIUM
UR28	Functional	Implement data anonymization techniques to protect sensitive information	HIGH
UR29	Functional	Develop a system for tracking data usage, sharing metrics, and maintaining version histories to ensure data traceability.	HIGH
UR30	Functional	Develop a centralized data repository for storing integrated health research data and implement data integration workflows to automate data ingestion from various sources	HIGH
UR31	Functional	Implement APIs for seamless data sharing with external partners	HIGH
UR32	Functional	Develop user-friendly data visualization tools	MEDIUM
UR33	Functional	Develop a comprehensive access control system to manage user permissions	HIGH
UR34	Functional	Define and adopt standardized formats for annotation labels	HIGH
UR35	Functional	Automate conversion of existing datasets format into standardized formats	MEDIUM
UR36	Functional	Enable seamless data exchange between systems and devices through easy-to-use APIs	MEDIUM

UR37	Functional	Validate incoming data against predefined schemas and standards	MEDIUM
UR38	Functional	Create a metadata repository for standardized formats and protocols	MEDIUM
UR39	Functional	Optimize edge computing for on-device data processing and filtering, eliminating the reliance on centralized systems	MEDIUM
UR40	Functional	Automate deletion of outdated and expired data	MEDIUM
UR41	Functional	Allow real-time processing of multimodal data	MEDIUM
UR42	Functional	Integrate AI-driven insights and analytics into resource planning	HIGH
UR43	Functional	Conduct real-time quality assessment of multimodal data	MEDIUM
UR44	Functional	Provide intuitive interface for semi-automated annotation processes	HIGH
UR45	Functional	Inject human knowledge during the annotation processes	HIGH
UR46	Functional	Provide feedback to annotator for refining AI-generated annotations	MEDIUM
UR47	Functional	Develop tools for real-time visualization of annotations	HIGH
UR48	Functional	Generate reports on the annotation quality and consistency of data	MEDIUM
UR49	Functional	Support metadata tagging in annotations to include context-specific information	HIGH
UR50	Functional	Provide secure data-sharing infrastructure with encryption mechanisms	HIGH
UR51	Functional	Enable datasets discoverability with AI-based brokering mechanisms and recommendation engines	HIGH
UR52	Functional	Implement role-based access control for data access	HIGH
UR53	Functional	Provide real-time anonymization of sensitive and personal data	HIGH
UR54	Functional	Define a standard data set of parameters to be logged for lifetime monitoring of BESS	MEDIUM
UR55	Functional	Develop and deploy necessary tools for collecting data based on defined standard dataset	HIGH
UR56	Functional	Share necessary BESS data via the dedicated PLIADES Data Space (e.g. lifetime monitoring data, CAD, datasheets, Bill of Materials, etc.)	HIGH

UR57	Functional	Manage data accessibility for different stakeholders within the platform	HIGH
UR58	Functional	Develop dashboards for visualizing battery data life cycles	HIGH
UR59	Functional	Develop analytics tools for life cycle data (e.g. battery state estimation and BESS predictive maintenance)	HIGH
UR60	Functional	Create accurate predictions of battery status with the development and training of AI/ML models	HIGH
UR61	Functional	Develop cross-domain AI models for federated applications	HIGH

Table 8 Non-functional user requirements of the PLIADES platform

ID	Type	Description	Priority Level
UR62	Non-Functional	Utilize historical data to anticipate production bottlenecks and adjust processes proactively	MEDIUM
UR63	Non-Functional	Generate alerts for deviations in material quality based on predefined thresholds	HIGH
UR64	Non-Functional	Establish data governance policies to ensure data quality and consistency	HIGH
UR65	Non-Functional	Enable secure data sharing protocols across stakeholders, maintaining data sovereignty	MEDIUM
UR66	Non-Functional	Develop data anonymization techniques for sharing insights without compromising sensitive information	HIGH
UR67	Non-Functional	Develop tools for tracking data usage and auditing trails for accountability	HIGH
UR68	Non-Functional	Use advanced encryption techniques for data stored on cloud servers	MEDIUM
UR69	Non-Functional	Establish a framework for continuous data quality monitoring and improvement	MEDIUM
UR70	Non-Functional	Enable data-driven insights for regulatory reporting	LOW
UR71	Non-Functional	Develop partnerships with technology companies to integrate DENNDATA into their solutions	HIGH
UR72	Non-Functional	Identify potential new sectors for DENNDATA expansion, such as healthcare or energy	MEDIUM
UR73	Non-Functional	Establish co-innovation programs with clients to develop new use cases for DENNDATA	MEDIUM

UR74	Non-Functional	Use predictive analytics to identify emerging market trends	MEDIUM
UR75	Non-Functional	Automate the generation of market reports based on data insights	HIGH
UR76	Non-Functional	Establish legal frameworks for co-development agreements with clients and partners	HIGH
UR77	Non-Functional	Define policies for sustainable data operations and archiving	MEDIUM
UR78	Non-Functional	Provide training programs on new data sharing protocols and tools	MEDIUM
UR79	Non-Functional	Develop training programs on data privacy and security protocols	MEDIUM
UR80	Non-Functional	Define and adopt standardized data formats	MEDIUM
UR81	Non-Functional	Define and adopt standardized metadata formats	MEDIUM
UR82	Non-Functional	Monitor resource usage to optimize data operations	MEDIUM
UR83	Non-Functional	Ensure effective and scalable data operations	MEDIUM
UR84	Non-Functional	Implement cloud-based data operations for cost effectiveness	MEDIUM
UR85	Non-Functional	Train annotators on the use of the automated tools	HIGH
UR86	Non-Functional	Define and establish numerical metrics to evaluate annotation quality and consistency	HIGH
UR87	Non-Functional	Ensure compliance with privacy regulations in data-sharing activities	HIGH
UR88	Non-Functional	Define data access policies to control data sharing	HIGH
UR89	Non-Functional	Manage data sharing agreements and permissions through user-friendly interfaces	HIGH
UR90	Non-Functional	Continuously monitor and log data-sharing activity	MEDIUM

12 Annex II: Priority Ranking of the System Functional Requirements

Table 9 System Functional Requirements

No. /ID	System Functional Requirement	Feasibility/ Relevance
SFR1	Collect raw data from multiple sources	HIGH
SFR2	Collect raw data from multiple agents	LOW
SFR3	Synchronize data based on timestamps	HIGH
SFR4	Synchronize data based on location	LOW
SFR5	Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics	HIGH
SFR6	Prioritize data based on their quality metrics	MEDIUM
SFR7	Apply harmonization pre-processing techniques to ensure data consistency	HIGH
SFR8	Perform ad-hoc analytics pre-processing and variable engineering methods	MEDIUM
SFR9	Generate annotation labels for data in a semi-automated manner	MEDIUM
SFR10	Extract automated labels for datasets assigned with confidence level	MEDIUM
SFR11	Allow human validation of dataset labels with low confidence	HIGH
SFR12	Inject human knowledge into data annotation and metadata creation processes	HIGH
SFR13	Annotate multimodal data	HIGH

SFR14	Visualize existing annotations of multimodal data	MEDIUM
SFR15	Convert data between different formats	HIGH
SFR16	Convert datasets into actionable data products	HIGH
SFR17	Establish automated monitoring and evaluation mechanisms for tracking AI model performance improvements	HIGH
SFR18	Interface to collect human feedback for improving AI performance	HIGH
SFR19	Create artificial data from raw data	MEDIUM
SFR20	Enable federated learning of AI models from heterogeneous data sources	HIGH
SFR21	Provide explanations to AI models' inference	MEDIUM
SFR22	Develop online re-training strategies for AI models	HIGH
SFR23	Apply cryptographic techniques during data exchange	HIGH
SFR24	Analyze and track data transactions	HIGH
SFR25	Anonymize or pseudonymize data by removing personally identifiable information	HIGH
SFR26	Ensure governance and rules in the control plane	HIGH
SFR27	Enforce compliance of data transactions according to legal and ownership restrictions	HIGH
SFR28	Store sensitive data with restrictions	HIGH
SFR29	Provide decentralized data connectors for secure, private, and sovereign data transfer between cloud and edge	HIGH
SFR30	Provide decentralized mechanisms for identity access management	HIGH

SFR31	Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors	HIGH
SFR32	Provide traceability and transparency for transactions in the entire network	HIGH
SFR33	Identify security vulnerabilities in data space architecture and protocols	HIGH
SFR34	Identify and provide alerts for suspicious activity and potential security threads	HIGH
SFR35	Analyze and prioritize threats within a dataspace	LOW
SFR36	Identify personal and non-personal data	HIGH
SFR37	Generate data flow maps with the entire lineage of each dataset	MEDIUM
SFR38	Identify potential privacy risks and threats	MEDIUM
SFR39	Create federated Digital Twin models	HIGH
SFR40	Enable cross-domain data discovery based on context and predefined criteria	MEDIUM
SFR41	Identify data drifts on multimodal data and provide alerts when detected	HIGH
SFR42	Provide visual analytics and decision-making capabilities	HIGH
SFR43	Extract similarity metrics for multimodal data	MEDIUM
SFR44	Discard data based on similarity metrics	LOW
SFR45	Perform automated data disposal processes based on expiration date	MEDIUM
SFR46	Allow data exchange with ontology-based components	HIGH
SFR47	Perform cross-domain adaptation between datasets	MEDIUM

SFR48	Store the ontology and data into a structured database	HIGH
SFR49	Allow queries to explore existing databases and internal relations	MEDIUM
SFR50	Provide strategies for data interoperability between data spaces	HIGH
SFR51	Receive natural language inputs to perform queries in Graphical Databases	MEDIUM
SFR52	Provide optimal recommendations between datasets and data consumer needs	MEDIUM
SFR53	Evaluate continuously the quality of data	HIGH
SFR54	Implement strategies to improve data quality	HIGH
SFR55	Extract decentralized analytics in real-time	HIGH
SFR56	Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing	HIGH

13 Annex III: Mapping of the User Requirements to Technical Specifications

Table 10 Summary of Mapping of User Requirements to Technical Specifications for SubUC1.1

	High Priority	Medium Priority	Low Priority
Technical specifications for SubUC1.1. Increment DENN data platform penetration in reluctant sectors through data compliance and security			
Related Requirements **Functional:** *UR1, UR2, UR3, UR4, UR5, UR6, UR7, UR8, UR9, UR10, UR11, UR12, UR13, UR14, UR15/* **Non-Functional:** *UR62, UR63, UR64, UR65, UR66, UR67, UR68, UR69, UR70, UR71, UR72, UR73, UR74, UR75, UR76*			
System Functional Requirements	**Performance requirements**	**Hardware requirements**	**Software requirements**
SFR16 - Convert datasets into actionable data products	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR23 - Apply cryptographic techniques during data exchange	Real-time	Basic CPU/GPU capabilities	Python
SFR24 - Analyze and track data transactions	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR26 - Ensure governance and rules in the control plane	Full assurance	Basic CPU/GPU capabilities	Python
SFR27 - Enforce compliance of data transactions according to legal and ownership restrictions	Full compliance	Basic CPU/GPU capabilities	Python, Go
SFR29 - Provide decentralized data connectors for secure, private, and sovereign data transfer between cloud and edge	N/A	Basic CPU/GPU capabilities	Python, Go
SFR30 - Provide decentralized mechanisms for identity access management	N/A	Basic CPU/GPU capabilities	Go

SFR31 - Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors	Near real-time	Basic CPU/GPU capabilities	Python, Go
SFR32 - Provide traceability and transparency for transactions in the entire network	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR33 - Identify security vulnerabilities in data space architecture and protocols	Accuracy > 95%	Basic CPU/GPU capabilities	Python, Go, C++
SFR37 - Generate data flow maps with the entire lineage of each dataset	N/A	Basic CPU/GPU capabilities	Python, C++
SFR38 - Identify potential privacy risks and threats	Accuracy > 90%	Basic CPU/GPU capabilities	Python, C++
SFR42 - Provide visual analytics and decision-making capabilities	Near real-time	Basic CPU/GPU capabilities, GUI-capable device	Python
SFR46 - Allow data exchange with ontology-based components	N/A	Basic CPU/GPU capabilities	Python, Go
SFR48 - Store the ontology and data into a structured database	N/A	Basic CPU/GPU capabilities	Python
SFR49 - Allow queries to explore existing databases and internal relations	Query processing time < 5s	Basic CPU/GPU capabilities	Python, Go
SFR52 - Provide optimal recommendations between datasets and data consumer needs	Recommendation accuracy > 90%	High GPU capabilities	C#, .NET, Python

Table 11 Summary of Mapping of User Requirements to Technical Specifications for SubUC1.2

	High Priority	Medium Priority	Low Priority
Technical specifications for SubUC1.2. Data-Driven Lifecycle Management for Suppliers			
Related Requirements **Functional:** *UR1, UR2, UR3, UR4, UR5, UR6, UR7, UR8, UR9, UR10, UR11, UR12, UR13, UR14, UR15* / **Non-Functional:** *UR62, UR63, UR64, UR65, UR66, UR67, UR68, UR69, UR70, UR71, UR72, UR73, UR74, UR75, UR76*			
System Functional Requirements	**Performance requirements**	**Hardware requirements**	**Software requirements**
SFR1 - Collect raw data from multiple sources	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, sensor APIs, hardware drivers
SFR3 - Synchronize data based on timestamps	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, access to NTP server
SFR5 - Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR6 - Prioritize data based on their quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR7 - Apply harmonization pre-processing techniques to ensure data consistency	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR8 - Perform ad-hoc analytics pre-processing and variable engineering methods	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR16 - Convert datasets into actionable data products	Near real-time	Basic CPU/GPU capabilities	Python, Rest API

SFR17 - Establish automated monitoring and evaluation mechanisms for tracking AI model performance improvements	Real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR22 - Develop online re-training strategies for AI models	TBD according to model types	High GPU capabilities	Python, Pytorch
SFR23 - Apply cryptographic techniques during data exchange	Real-time	Basic CPU/GPU capabilities	Python
SFR26 - Ensure governance and rules in the control plane	Full assurance	Basic CPU/GPU capabilities	Python
SFR27 - Enforce compliance of data transactions according to legal and ownership restrictions	Full compliance	Basic CPU/GPU capabilities	Python, Go
SFR29 - Provide decentralized data connectors for secure, private, and sovereign data transfer between cloud and edge	N/A	Basic CPU/GPU capabilities	Python, Go
SFR30 - Provide decentralized mechanisms for identity access management	N/A	Basic CPU/GPU capabilities	Go
SFR31 - Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors	Near real-time	Basic CPU/GPU capabilities	Python, Go
SFR32 - Provide traceability and transparency for transactions in the entire network	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR33 - Identify security vulnerabilities in data space architecture and protocols	Accuracy > 95%	Basic CPU/GPU capabilities	Python, Go, C++
SFR37 - Generate data flow maps with the entire lineage of each dataset	N/A	Basic CPU/GPU capabilities	Python, C++
SFR38 - Identify potential privacy risks and threats	Accuracy > 90%	Basic CPU/GPU capabilities	Python, C++

SFR42 - Provide visual analytics and decision-making capabilities	Near real-time	Basic CPU/GPU capabilities, GUI-capable device	Python
SFR53 - Evaluate continuously the quality of data	Real-time	High GPU capabilities	Python
SFR55 - Extract decentralized analytics in real-time	Real-time	High GPU capabilities	Python, C++
SFR56 - Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing	N/A	High GPU capabilities	C#, .NET, Python

Table 12 Summary of Mapping of User Requirements to Technical Specifications for SubUC1.3

	High Priority	**Medium Priority**	**Low Priority**
Technical specifications for SubUC1.3. Commercial Expansion of DENNDATA for Cross-Industry Data-Driven Optimization			
Related Requirements **Functional:** *UR1, UR2, UR3, UR4, UR5, UR6, UR7, UR8, UR9, UR10, UR11, UR12, UR13, UR14, UR15* / **Non-Functional:** *UR62, UR63, UR64, UR65, UR66, UR67, UR68, UR69, UR70, UR71, UR72, UR73, UR74, UR75, UR76*			
System Functional Requirements	**Performance requirements**	**Hardware requirements**	**Software requirements**
SFR1 - Collect raw data from multiple sources	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, sensor APIs, hardware drivers
SFR3 - Synchronize data based on timestamps	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, access to NTP server

SFR5 - Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR7 - Apply harmonization pre-processing techniques to ensure data consistency	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR8 - Perform ad-hoc analytics pre-processing and variable engineering methods	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR16 - Convert datasets into actionable data products	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR23 - Apply cryptographic techniques during data exchange	Real-time	Basic CPU/GPU capabilities	Python
SFR24 - Analyze and track data transactions	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR26 - Ensure governance and rules in the control plane	Full assurance	Basic CPU/GPU capabilities	Python
SFR27 - Enforce compliance of data transactions according to legal and ownership restrictions	Full compliance	Basic CPU/GPU capabilities	Python, Go
SFR29 - Provide decentralized data connectors for secure, private, and sovereign data transfer between cloud and edge	N/A	Basic CPU/GPU capabilities	Python, Go
SFR30 - Provide decentralized mechanisms for identity access management	N/A	Basic CPU/GPU capabilities	Go
SFR31 - Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors	Near real-time	Basic CPU/GPU capabilities	Python, Go
SFR32 - Provide traceability and transparency for transactions in the entire network	Real-time	Basic CPU/GPU capabilities	Python, Go

SFR33 - Identify security vulnerabilities in data space architecture and protocols	Accuracy > 95%	Basic CPU/GPU capabilities	Python, Go, C++
SFR35 - Analyze and prioritize threats within a dataspace	Real-time	Basic CPU/GPU capabilities	Python
SFR37 - Generate data flow maps with the entire lineage of each dataset	N/A	Basic CPU/GPU capabilities	Python, C++
SFR42 - Provide visual analytics and decision-making capabilities	Near real-time	Basic CPU/GPU capabilities, GUI-capable device	Python
SFR55 - Extract decentralized analytics in real-time	Real-time	High GPU capabilities	Python, C++
SFR56 - Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing	N/A	High GPU capabilities	C#, .NET, Python

Table 13 Summary of Mapping of User Requirements to Technical Specifications for SubUC2.1 and SubUC2.2

	High Priority	**Medium Priority**	**Low Priority**
Technical specifications for SubUC2.1. HRI in rehabilitation and SubUC2.2. HRI in patient monitoring			
Related Requirements **Functional:** *UR16*, *UR17*, *UR18*, *UR19*, *UR20*, *UR21*, *UR22*, *UR23*, *UR24*, *UR25*, *UR26*, *UR27*, *UR28*, *UR29* / **Non-Functional:** *UR78*			
System Functional Requirements	**Performance requirements**	**Hardware requirements**	**Software requirements**

SFR1 - Collect raw data from multiple sources	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, sensor APIs, hardware drivers
SFR3 - Synchronize data based on timestamps	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, access to NTP server
SFR5 - Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR6 - Prioritize data based on their quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR7 - Apply harmonization pre-processing techniques to ensure data consistency	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR9 - Generate annotation labels for data in a semi-automated manner	>90% confidence	Increased GPU capabilities	Python, Pytorch
SFR10 - Extract automated labels for datasets assigned with confidence level	>90% confidence	Increased GPU capabilities	Python, Pytorch
SFR11 - Allow human validation of dataset labels with low confidence	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser
SFR12 - Inject human knowledge into data annotation and metadata creation processes	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR13 - Annotate multimodal data	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR14 - Visualize existing annotations of multimodal data	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular

SFR17 - Establish automated monitoring and evaluation mechanisms for tracking AI model performance improvements	Real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR18 - Interface to collect human feedback for improving AI performance	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR19 - Create artificial data from raw data	Similarity to real data > 90%	High GPU capabilities	Python, Pytorch, Simulation engines
SFR20 - Enable federated learning of AI models from heterogeneous data sources	TBD according to model types	High GPU capabilities	Python, REST API
SFR21 - Provide explanations to AI models' inference	TBD according to model types	High GPU capabilities	Python, Pytorch
SFR22 - Develop online re-training strategies for AI models	TBD according to model types	High GPU capabilities	Python, Pytorch
SFR25 - Anonymize or pseudonymize data by removing personally identifiable information	Accuracy >95%	Basic CPU/GPU capabilities	Python, Go
SFR28 - Store sensitive data with restrictions	N/A	Basic CPU/GPU capabilities	Python, Go
SFR29 - Provide decentralized data connectors for secure, private, and sovereign data transfer between cloud and edge	N/A	Basic CPU/GPU capabilities	Python, Go
SFR31 - Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors	Near real-time	Basic CPU/GPU capabilities	Python, Go
SFR32 - Provide traceability and transparency for transactions in the entire network	Real-time	Basic CPU/GPU capabilities	Python, Go

SFR33 - Identify security vulnerabilities in data space architecture and protocols	Accuracy > 95%	Basic CPU/GPU capabilities	Python, Go, C++
SFR34 - Identify and provide alerts for suspicious activity and potential security threads	Accuracy > 95%	Basic CPU/GPU capabilities	Python, Go
SFR35 - Analyze and prioritize threats within a dataspace	Real-time	Basic CPU/GPU capabilities	Python
SFR36 - Identify personal and non-personal data	Accuracy > 90%	Basic CPU/GPU capabilities	Python
SFR38 - Identify potential privacy risks and threats	Accuracy > 90%	Basic CPU/GPU capabilities	Python, C++
SFR39 - Create federated Digital Twin models	N/A	High GPU capabilities	Python, Pytorch, Simulation engines
SFR41 - Identify data drifts on multimodal data and provide alerts when detected	Accuracy > 95%	High GPU capabilities	Python
SFR43 - Extract similarity metrics for multimodal data	Near real-time	High GPU capabilities	Python
SFR44 - Discard data based on similarity metrics	Real-time	Basic CPU/GPU capabilities	Python
SFR45 - Perform automated data disposal processes based on expiration date	Near real-time	Basic CPU/GPU capabilities	Python
SFR51 - Receive natural language inputs to perform queries in Graphical Databases	Process query time < 1s	High GPU capabilities	Python
SFR53 - Evaluate continuously the quality of data	Real-time	High GPU capabilities	Python
SFR54 - Implement strategies to improve data quality	N/A	Basic CPU/GPU capabilities	Python

SFR56 - Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing	N/A	High GPU capabilities	C#, .NET, Python

Table 14 Summary of Mapping of User Requirements to Technical Specifications for SubUC3.1 and SubUC3.2

	High Priority	**Medium Priority**	**Low Priority**
Technical specifications for SubUC3.1. Sharing Data and Models among Partners and SubUC3.2. Collaborative Data/Service Exchange and Third-Party Data Acquisition			
Related Requirements **Functional:** *UR26*, *UR28*, *UR29*, *UR30*, *UR31*, *UR32*, *UR33* / **Non-Functional:** *UR66*, *UR78*, *UR79*			
System Functional Requirements	**Performance requirements**	**Hardware requirements**	**Software requirements**
SFR1 - Collect raw data from multiple sources	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, sensor APIs, hardware drivers
SFR5 - Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR8 - Perform ad-hoc analytics pre-processing and variable engineering methods	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR9 - Generate annotation labels for data in a semi-automated manner	>90% confidence	Increased GPU capabilities	Python, Pytorch

SFR12 - Inject human knowledge into data annotation and metadata creation processes	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR13 - Annotate multimodal data	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR16 - Convert datasets into actionable data products	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR17 - Establish automated monitoring and evaluation mechanisms for tracking AI model performance improvements	Real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR20 - Enable federated learning of AI models from heterogeneous data sources	TBD according to model types	High GPU capabilities	Python, REST API
SFR21 - Provide explanations to AI models' inference	TBD according to model types	High GPU capabilities	Python, Pytorch
SFR23 - Apply cryptographic techniques during data exchange	Real-time	Basic CPU/GPU capabilities	Python
SFR24 - Analyze and track data transactions	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR25 - Anonymize or pseudonymize data by removing personally identifiable information	Accuracy >95%	Basic CPU/GPU capabilities	Python, Go
SFR27 - Enforce compliance of data transactions according to legal and ownership restrictions	Full compliance	Basic CPU/GPU capabilities	Python, Go
SFR28 - Store sensitive data with restrictions	N/A	Basic CPU/GPU capabilities	Python, Go
SFR31 - Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors	Near real-time	Basic CPU/GPU capabilities	Python, Go

SFR32 - Provide traceability and transparency for transactions in the entire network	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR33 - Identify security vulnerabilities in data space architecture and protocols	Accuracy > 95%	Basic CPU/GPU capabilities	Python, Go, C++
SFR36 - Identify personal and non-personal data	Accuracy > 90%	Basic CPU/GPU capabilities	Python
SFR41 - Identify data drifts on multimodal data and provide alerts when detected	Accuracy > 95%	High GPU capabilities	Python
SFR43 - Extract similarity metrics for multimodal data	Near real-time	High GPU capabilities	Python
SFR44 - Discard data based on similarity metrics	Real-time	Basic CPU/GPU capabilities	Python
SFR45 - Perform automated data disposal processes based on expiration date	Near real-time	Basic CPU/GPU capabilities	Python
SFR46 - Allow data exchange with ontology-based components	N/A	Basic CPU/GPU capabilities	Python, Go
SFR48 - Store the ontology and data into a structured database	N/A	Basic CPU/GPU capabilities	Python
SFR50 - Provide strategies for data interoperability between data spaces	N/A	Basic CPU/GPU capabilities	Python
SFR53 - Evaluate continuously the quality of data	Real-time	High GPU capabilities	Python
SFR54 - Implement strategies to improve data quality	N/A	Basic CPU/GPU capabilities	Python
SFR55 - Extract decentralized analytics in real-time	Real-time	High GPU capabilities	Python, C++
SFR56 - Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing	N/A	High GPU capabilities	C#, .NET, Python

Table 15 Summary of Mapping of User Requirements to Technical Specifications for SubUC4.1

	High Priority	Medium Priority	Low Priority

Technical specifications for SubUC4.1: AI-based ADAS development

Related Requirements

Functional: *UR16, UR18, UR25, UR27, UR34, UR35, UR36, UR37, UR38, UR40, UR41, UR42, UR43, UR44, UR45, UR46, UR47, UR48, UR49, UR50, UR51, UR52, UR53*

Non-Functional: *UR81, UR82, UR83, UR84, UR85, UR86, UR87, UR88, UR89, UR90*

System Functional Requirements	Performance requirements	Hardware requirements	Software requirements
SFR1 - Collect raw data from multiple sources	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, sensor APIs, hardware drivers
SFR3 - Synchronize data based on timestamps	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, access to NTP server
SFR5 - Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR6 - Prioritize data based on their quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR7 - Apply harmonization pre-processing techniques to ensure data consistency	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR9 - Generate annotation labels for data in a semi-automated manner	>90% confidence	Increased GPU capabilities	Python, Pytorch

SFR11 - Allow human validation of dataset labels with low confidence	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser
SFR12 - Inject human knowledge into data annotation and metadata creation processes	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR13 - Annotate multimodal data	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR14 - Visualize existing annotations of multimodal data	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR15 - Convert data between different formats	Real-time	Basic CPU/GPU capabilities	Python, REST API
SFR18 - Interface to collect human feedback for improving AI performance	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR19 - Create artificial data from raw data	Similarity to real data > 90%	High GPU capabilities	Python, Pytorch, Simulation engines
SFR20 - Enable federated learning of AI models from heterogeneous data sources	TBD according to model types	High GPU capabilities	Python, REST API
SFR21 - Provide explanations to AI models’ inference	TBD according to model types	High GPU capabilities	Python, Pytorch
SFR22 - Develop online re-training strategies for AI models	TBD according to model types	High GPU capabilities	Python, Pytorch

SFR24 - Analyze and track data transactions	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR25 - Anonymize or pseudonymize data by removing personally identifiable information	Accuracy >95%	Basic CPU/GPU capabilities	Python, Go
SFR26 - Ensure governance and rules in the control plane	Full assurance	Basic CPU/GPU capabilities	Python
SFR27 - Enforce compliance of data transactions according to legal and ownership restrictions	Full compliance	Basic CPU/GPU capabilities	Python, Go
SFR28 - Store sensitive data with restrictions	N/A	Basic CPU/GPU capabilities	Python, Go
SFR29 - Provide decentralized data connectors for secure, private, and sovereign data transfer between cloud and edge	N/A	Basic CPU/GPU capabilities	Python, Go
SFR31 - Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors	Near real-time	Basic CPU/GPU capabilities	Python, Go
SFR32 - Provide traceability and transparency for transactions in the entire network	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR33 - Identify security vulnerabilities in data space architecture and protocols	Accuracy > 95%	Basic CPU/GPU capabilities	Python, Go, C++
SFR35 - Analyze and prioritize threats within a dataspace	Real-time	Basic CPU/GPU capabilities	Python
SFR36 - Identify personal and non-personal data	Accuracy > 90%	Basic CPU/GPU capabilities	Python
SFR37 - Generate data flow maps with the entire lineage of each dataset	N/A	Basic CPU/GPU capabilities	Python, C++
SFR39 - Create federated Digital Twin models	N/A	High GPU capabilities	Python, Pytorch, Simulation engines

SFR40 - Enable cross-domain data discovery based on context and predefined criteria	Accuracy of suggestions > 90%	High GPU capabilities	C#, .NET, Python
SFR42 - Provide visual analytics and decision-making capabilities	Near real-time	Basic CPU/GPU capabilities, GUI-capable device	Python
SFR43 - Extract similarity metrics for multimodal data	Near real-time	High GPU capabilities	Python
SFR44 - Discard data based on similarity metrics	Real-time	Basic CPU/GPU capabilities	Python
SFR46 - Allow data exchange with ontology-based components	N/A	Basic CPU/GPU capabilities	Python, Go
SFR47 - Perform cross-domain adaptation between datasets	Depends on the dataset	High GPU capabilities	Python
SFR49 - Allow queries to explore existing databases and internal relations	Query processing time < 5s	Basic CPU/GPU capabilities	Python, Go
SFR50 - Provide strategies for data interoperability between data spaces	N/A	Basic CPU/GPU capabilities	Python
SFR51 - Receive natural language inputs to perform queries in Graphical Databases	Process query time < 1s	High GPU capabilities	Python
SFR52 - Provide optimal recommendations between datasets and data consumer needs	Recommendation accuracy > 90%	High GPU capabilities	C#, .NET, Python
SFR53 - Evaluate continuously the quality of data	Real-time	High GPU capabilities	Python
SFR54 - Implement strategies to improve data quality	N/A	Basic CPU/GPU capabilities	Python
SFR55 - Extract decentralized analytics in real-time	Real-time	High GPU capabilities	Python, C++

SFR56 - Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing	N/A	High GPU capabilities	C#, .NET, Python

Table 16 Summary of Mapping of User Requirements to Technical Specifications for SubUC4.2

	High Priority	**Medium Priority**	**Low Priority**
Technical specifications for SubUC4.2. AI-based traffic management			
Related Requirements **Functional:** *UR16, UR18, UR25, UR27, UR34, UR35, UR36, UR37, UR41, UR42, UR43, UR44, UR45, UR46, UR47, UR48, UR49, UR50, UR51, UR52, UR53* / **Non-Functional:** *UR77, UR80, UR81, UR82, UR83, UR84, UR85, UR86, UR87, UR88, UR89*			
System Functional Requirements	**Performance requirements**	**Hardware requirements**	**Software requirements**
SFR1 - Collect raw data from multiple sources	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, sensor APIs, hardware drivers
SFR2 - Collect raw data from multiple agents	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, sensor APIs, hardware drivers, network interfaces
SFR3 - Synchronize data based on timestamps	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, access to NTP server
SFR4 - Synchronize data based on location	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, access to location information
SFR5 - Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API

SFR6 - Prioritize data based on their quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR7 - Apply harmonization pre-processing techniques to ensure data consistency	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR9 - Generate annotation labels for data in a semi-automated manner	>90% confidence	Increased GPU capabilities	Python, Pytorch
SFR11 - Allow human validation of dataset labels with low confidence	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser
SFR12 - Inject human knowledge into data annotation and metadata creation processes	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR13 - Annotate multimodal data	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR14 - Visualize existing annotations of multimodal data	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR15 - Convert data between different formats	Real-time	Basic CPU/GPU capabilities	Python, REST API
SFR18 - Interface to collect human feedback for improving AI performance	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR19 - Create artificial data from raw data	Similarity to real data > 90%	High GPU capabilities	Python, Pytorch, Simulation engines
SFR20 - Enable federated learning of AI models from heterogeneous data sources	TBD according to model types	High GPU capabilities	Python, REST API
SFR21 - Provide explanations to AI models’ inference	TBD according to model types	High GPU capabilities	Python, Pytorch

SFR22 - Develop online re-training strategies for AI models	TBD according to model types	High GPU capabilities	Python, Pytorch
SFR24 - Analyze and track data transactions	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR25 - Anonymize or pseudonymize data by removing personally identifiable information	Accuracy >95%	Basic CPU/GPU capabilities	Python, Go
SFR26 - Ensure governance and rules in the control plane	Full assurance	Basic CPU/GPU capabilities	Python
SFR27 - Enforce compliance of data transactions according to legal and ownership restrictions	Full compliance	Basic CPU/GPU capabilities	Python, Go
SFR28 - Store sensitive data with restrictions	N/A	Basic CPU/GPU capabilities	Python, Go
SFR29 - Provide decentralized data connectors for secure, private, and sovereign data transfer between cloud and edge	N/A	Basic CPU/GPU capabilities	Python, Go
SFR31 - Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors	Near real-time	Basic CPU/GPU capabilities	Python, Go
SFR32 - Provide traceability and transparency for transactions in the entire network	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR33 - Identify security vulnerabilities in data space architecture and protocols	Accuracy > 95%	Basic CPU/GPU capabilities	Python, Go, C++
SFR35 - Analyze and prioritize threats within a dataspace	Real-time	Basic CPU/GPU capabilities	Python
SFR36 - Identify personal and non-personal data	Accuracy > 90%	Basic CPU/GPU capabilities	Python
SFR37 - Generate data flow maps with the entire lineage of each dataset	N/A	Basic CPU/GPU capabilities	Python, C++

SFR39 - Create federated Digital Twin models	N/A	High GPU capabilities	Python, Pytorch, Simulation engines
SFR42 - Provide visual analytics and decision-making capabilities	Near real-time	Basic CPU/GPU capabilities, GUI-capable device	Python
SFR43 - Extract similarity metrics for multimodal data	Near real-time	High GPU capabilities	Python
SFR44 - Discard data based on similarity metrics	Real-time	Basic CPU/GPU capabilities	Python
SFR46 - Allow data exchange with ontology-based components	N/A	Basic CPU/GPU capabilities	Python, Go
SFR49 - Allow queries to explore existing databases and internal relations	Query processing time < 5s	Basic CPU/GPU capabilities	Python, Go
SFR50 - Provide strategies for data interoperability between data spaces	N/A	Basic CPU/GPU capabilities	Python
SFR51 - Receive natural language inputs to perform queries in Graphical Databases	Process query time < 1s	High GPU capabilities	Python
SFR52 - Provide optimal recommendations between datasets and data consumer needs	Recommendation accuracy > 90%	High GPU capabilities	C#, .NET, Python
SFR53 - Evaluate continuously the quality of data	Real-time	High GPU capabilities	Python
SFR54 - Implement strategies to improve data quality	N/A	Basic CPU/GPU capabilities	Python
SFR56 - Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing	N/A	High GPU capabilities	C#, .NET, Python

Table 17 Summary of Mapping of User Requirements to Technical Specifications for SubUC5.1

	High Priority	Medium Priority	Low Priority
Technical specifications for SubUC5.1. Sharing field data from battery systems energy storage installations			
Related Requirements **Functional:** *UR54*, *UR55*, *UR56*, *UR57*, *UR58*, *UR59*, *UR60*, *UR61*			
System Functional Requirements	**Performance requirements**	**Hardware requirements**	**Software requirements**
SFR1 - Collect raw data from multiple sources	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, sensor APIs, hardware drivers
SFR5 - Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR7 - Apply harmonization pre-processing techniques to ensure data consistency	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR8 - Perform ad-hoc analytics pre-processing and variable engineering methods	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR9 - Generate annotation labels for data in a semi-automated manner	>90% confidence	Increased GPU capabilities	Python, Pytorch
SFR10 - Extract automated labels for datasets assigned with confidence level	>90% confidence	Increased GPU capabilities	Python, Pytorch
SFR16 - Convert datasets into actionable data products	Near real-time	Basic CPU/GPU capabilities	Python, Rest API

SFR17 - Establish automated monitoring and evaluation mechanisms for tracking AI model performance improvements	Real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR19 - Create artificial data from raw data	Similarity to real data > 90%	High GPU capabilities	Python, Pytorch, Simulation engines
SFR22 - Develop online re-training strategies for AI models	TBD according to model types	High GPU capabilities	Python, Pytorch
SFR23 - Apply cryptographic techniques during data exchange	Real-time	Basic CPU/GPU capabilities	Python
SFR26 - Ensure governance and rules in the control plane	Full assurance	Basic CPU/GPU capabilities	Python
SFR27 - Enforce compliance of data transactions according to legal and ownership restrictions	Full compliance	Basic CPU/GPU capabilities	Python, Go
SFR30 - Provide decentralized mechanisms for identity access management	N/A	Basic CPU/GPU capabilities	Go
SFR31 - Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors	Near real-time	Basic CPU/GPU capabilities	Python, Go
SFR32 - Provide traceability and transparency for transactions in the entire network	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR33 - Identify security vulnerabilities in data space architecture and protocols	Accuracy > 95%	Basic CPU/GPU capabilities	Python, Go, C++
SFR37 - Generate data flow maps with the entire lineage of each dataset	N/A	Basic CPU/GPU capabilities	Python, C++
SFR39 - Create federated Digital Twin models	N/A	High GPU capabilities	Python, Pytorch, Simulation engines

SFR40 - Enable cross-domain data discovery based on context and predefined criteria	Accuracy of suggestions > 90%	High GPU capabilities	C#, .NET, Python
SFR42 - Provide visual analytics and decision-making capabilities	Near real-time	Basic CPU/GPU capabilities, GUI-capable device	Python
SFR45 - Perform automated data disposal processes based on expiration date	Near real-time	Basic CPU/GPU capabilities	Python
SFR47 - Perform cross-domain adaptation between datasets	Depends on the dataset	High GPU capabilities	Python
SFR48 - Store the ontology and data into a structured database	N/A	Basic CPU/GPU capabilities	Python
SFR50 - Provide strategies for data interoperability between data spaces	N/A	Basic CPU/GPU capabilities	Python
SFR55 - Extract decentralized analytics in real-time	Real-time	High GPU capabilities	Python, C++
SFR56 - Provide AI-driven connectors for data transformation, security and privacy- preserving data sharing	N/A	High GPU capabilities	C#, .NET, Python

Table 18 Summary of Mapping of User Requirements to Technical Specifications for SubUC6.1, SubUC6.2, and SubUC6.3

	High Priority	Medium Priority	Low Priority
Technical specifications for SubUC6.1. HRI in telepresence robot operation, SubUC6.2. HRI in rehabilitation robot operation, and SubUC6.3. HRI in manufacturing inspection robot operation			
Related Requirements			

Functional: *UR16*, *UR17*, *UR18*, *UR19*, *UR20*, *UR21*, *UR22*, *UR23*, *UR24*, *UR25*, *UR26*, *UR27*, *UR28*, *UR29* /

Non-Functional: *UR77*

System Functional Requirements	Performance requirements	Hardware requirements	Software requirements
SFR1 - Collect raw data from multiple sources	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, sensor APIs, hardware drivers
SFR2 - Collect raw data from multiple agents	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, sensor APIs, hardware drivers, network interfaces
SFR3 - Synchronize data based on timestamps	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, access to NTP server
SFR4 - Synchronize data based on location	Real-time	Basic CPU/GPU capabilities, High storage capacity	Python, access to location information
SFR5 - Apply filtering mechanisms on data to remove noise, duplicates, redundant information based on quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR6 - Prioritize data based on their quality metrics	Near real-time	Basic CPU/GPU capabilities	Python, Rest API
SFR9 - Generate annotation labels for data in a semi-automated manner	>90% confidence	Increased GPU capabilities	Python, Pytorch
SFR11 - Allow human validation of dataset labels with low confidence	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser
SFR12 - Inject human knowledge into data annotation and metadata creation processes	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular

SFR13 - Annotate multimodal data	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR14 - Visualize existing annotations of multimodal data	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR15 - Convert data between different formats	Real-time	Basic CPU/GPU capabilities	Python, REST API
SFR18 - Interface to collect human feedback for improving AI performance	Real-time	Basic CPU/GPU capabilities, GUI-capable device	Python, Web browser, Typescript, Angular
SFR19 - Create artificial data from raw data	Similarity to real data > 90%	High GPU capabilities	Python, Pytorch, Simulation engines
SFR20 - Enable federated learning of AI models from heterogeneous data sources	TBD according to model types	High GPU capabilities	Python, REST API
SFR21 - Provide explanations to AI models' inference	TBD according to model types	High GPU capabilities	Python, Pytorch
SFR22 - Develop online re-training strategies for AI models	TBD according to model types	High GPU capabilities	Python, Pytorch
SFR23 - Apply cryptographic techniques during data exchange	Real-time	Basic CPU/GPU capabilities	Python
SFR25 - Anonymize or pseudonymize data by removing personally identifiable information	Accuracy >95%	Basic CPU/GPU capabilities	Python, Go
SFR28 - Store sensitive data with restrictions	N/A	Basic CPU/GPU capabilities	Python, Go
SFR31 - Monitor and audit data transactions for compliance with legal frameworks and smart contracts of IDS connectors	Near real-time	Basic CPU/GPU capabilities	Python, Go

SFR32 - Provide traceability and transparency for transactions in the entire network	Real-time	Basic CPU/GPU capabilities	Python, Go
SFR33 - Identify security vulnerabilities in data space architecture and protocols	Accuracy > 95%	Basic CPU/GPU capabilities	Python, Go, C++
SFR34 - Identify and provide alerts for suspicious activity and potential security threads	Accuracy > 95%	Basic CPU/GPU capabilities	Python, Go
SFR35 - Analyze and prioritize threats within a dataspace	Real-time	Basic CPU/GPU capabilities	Python
SFR36 - Identify personal and non-personal data	Accuracy > 90%	Basic CPU/GPU capabilities	Python
SFR37 - Generate data flow maps with the entire lineage of each dataset	N/A	Basic CPU/GPU capabilities	Python, C++
SFR38 - Identify potential privacy risks and threats	Accuracy > 90%	Basic CPU/GPU capabilities	Python, C++
SFR40 - Enable cross-domain data discovery based on context and predefined criteria	Accuracy of suggestions > 90%	High GPU capabilities	C#, .NET, Python
SFR46 - Allow data exchange with ontology-based components	N/A	Basic CPU/GPU capabilities	Python, Go
SFR47 - Perform cross-domain adaptation between datasets	Depends on the dataset	High GPU capabilities	Python
SFR48 - Store the ontology and data into a structured database	N/A	Basic CPU/GPU capabilities	Python
SFR50 - Provide strategies for data interoperability between data spaces	N/A	Basic CPU/GPU capabilities	Python

SFR56 - Provide AI-driven connectors for data transformation, security and privacy-preserving data sharing	N/A	High GPU capabilities	C#, .NET, Python