Reliable AI and Data Optimization

D2.4 - Final RAIDO Platform and Architecture

Submission date: 30/06/2026

Project Number:	101135800		
Project Acronym:	RAIDO		
Project Title:	Reliable AI and Data Optimization		
Start date:	January 1st, 2024	**Duration:**	36 months
Deliverable:	D2.4 - Final RAIDO Platform and Architecture		
Work Package:	WP2		
Lead partner:	UBITECH-LIM (UBI)		
Author(s):	Dr. Christos Kotronis (UBI), Giorgos Theodorou (UBI), Nikos Androulidakis (8BELLS), Dimitrios-Christos Asimopoulos (MINDS), Giola Genni (MINDS), Nikolaos Ntampakis (MINDS), Georgios Tziolas (MINDS), Georgios Tsiamitas (MINDS), Ioannis Theologitis (CERTH), Nikos Kamadanis (CERTH), Stylianos Eleftheriadis (CERTH), Vasileios Lolis (CERTH)		
Reviewers:	QMUL		
Due date:	30/06/2026		
Deliverable Type:	R-Document, report	**Dissemination Level:**	PU-Public
Version number:	1.0		

Document History

Version	Date	Author	Description
0.1	03/06/2026	Dr. Christos Kotronis (UBI)	First release of Table of Contents (ToC).
0.2	05/06/2026	Dr. Christos Kotronis (UBI), Georgios Theodorou (UBI)	First draft of Executive Summary, Introduction, RTM, and Conclusions sections.
0.3	09/06/2026	Dr. Christos Kotronis (UBI), Nikos Androulidakis (8BELLS)	Incorporation of contributions from technical partners.
0.4	19/06/2026	Dimitrios-Christos Asimopoulos (MINDS), Giola Genni (MINDS), Nikolaos Ntampakis (MINDS), Georgios Tziolas (MINDS), Georgios Tsiamitas (MINDS)	Incorporation of contribution from MINDS.
0.5	22/6/2026	Ioannis Theologitis (CERTH), Nikos Kamadanis (CERTH), Stylianos Eleftheriadis (CERTH), Vasileios Lolis (CERTH)	Incorporation of contribution from CERTH.
0.6	24/06/2026	Dr. Christos Kotronis (UBI)	Preparation of final manuscript for internal review and quality control.
0.7	28/06/2026	Debin Meng (QMUL)	Internal review and recommendations for improvement.
1.0	30/06/2026	Dr. Christos Kotronis (UBI)	Preparation of final version and submission of the deliverable.

Table of Contents

List of Figures

List of Tables

Abbreviations

AE	AI Engineer
AI	Artificial Intelligence
API	Application Programming Interface
BLC	Blockchain and Smart Contracts
CAM	Class Activation Mapping
D	Deliverable
DATALAKE	RAIDO Data Lake
DATAGEN	Synthetic Data Generation
DC	Data Consumer
DE	Data Engineer
DISTIL	Few- and Zero-Shot Adaptation and Distillation of Vision-Language Models
DP	Data Provider
DT	Digital Twin
E2C	Edge-to-Cloud
EC	European Commission
ENRICH	Data Enrichment, Curation, Distillation and Federated Mining
EU	European Union
FEDCL	Federated and Continual Learning
FR	Functional Requirement
GAN	Generative Adversarial Network
GDPR	General Data Protection Regulation
GMUI	Green Monitoring User Interface
GRAD-CAM	Gradient-weighted Class Activation Mapping
HITL	Human-in-the-Loop
IPFS	InterPlanetary File System
JSON	JavaScript Object Notation
JWT	JSON Web Token
KPI	Key Performance Indicator
LL	Lifelong Learning
LRP	Layer-wise Relevance Propagation
M	Month
ML	Machine Learning
NFR	Non-Functional Requirement
PA	Platform/System Administrator
RAIDO	Reliable AI and Data Optimization
RBAC	Role-Based Access Control
REST	Representational State Transfer
RLFEED	Reinforced Benchmarking and Feedback-Based Progress Monitoring
RSCMANAGER	AI-based Edge-to-Cloud Continuum Resource Allocation and Deployment
RTM	Requirements Traceability Matrix
SDN	Software-Defined Networking
SHAP	SHapley Additive exPlanations
SLO	Service Level Objective
SOTA	State-of-the-Art
SP	System Property
T	Task
TrL	Trust Level
UI	Adaptive AI Processes and Visualization Engine (User Interface)
VAE	Variational Autoencoder
V-L	Vision-Language
VLM	Vision-Language Model
WP	Work Package
XAI	Explainable Artificial Intelligence

Disclaimer

Executive summary

The present Deliverable (D) 2.4 “Final RAIDO Platform and Architecture”, presents the final architectural specification and implementation status of the RAIDO platform developed throughout the project. Building upon the initial architectural design introduced in D2.3, this deliverable documents the evolution of the platform into a fully integrated, operational, and validated ecosystem, consolidating the technical developments, component integrations, deployment and validation activities, and architectural refinements carried out across Work Packages (WPs) 3, 4, 5, and 6.

RAIDO delivers a complete framework for trustworthy, explainable, and energy-efficient Artificial Intelligence (AI), supporting the entire AI lifecycle from data ingestion and management to model training, optimization, deployment, monitoring, and continuous improvement. The platform integrates advanced data optimization techniques, synthetic data generation services, federated and continual learning mechanisms, explainable AI frameworks, orchestration services, Edge-to-Cloud (E2C) resource management capabilities, benchmarking mechanisms, and blockchain-based traceability services into a unified ecosystem designed to support heterogeneous Industry 4.0 environments.

The final platform consists of thirteen (13) tightly integrated components that collectively provide data management (data ingestion, enrichment, synthetic data generation), AI model optimization, sustainable deployment, monitoring, explainability, benchmarking, and governance capabilities. These components are supported by a polyglot Data Lake infrastructure, a Kubernetes-based deployment environment, a secure backend service layer, and a centralized authentication and authorization framework, providing a scalable, modular, and interoperable environment for smart energy grids, smart farming and fungal food production, healthcare and pharmacogenomics, and robotics and fibre-characterization applications. The architecture has been successfully implemented, integrated, deployed, and validated through the RAIDO pilot activities, demonstrating interoperable end-to-end AI workflows across different application domains and distributed cloud, edge, and hybrid computing environments. The successful integration of the platform components confirms the feasibility of executing the complete data and AI lifecycle within a common operational ecosystem while supporting transparency, traceability, energy efficiency, and continuous optimization.

A major outcome of the RAIDO project is the realization of an operational platform integrating MySQL, MongoDB, MinIO, InfluxDB, Prometheus, Kepler, Apache Airflow, Hyperledger Fabric, InterPlanetary File System (IPFS), Keycloak, and Kubernetes technologies into a cohesive architecture. Through a comprehensive set of Representational State Transfer (REST) Application Programming Interfaces (APIs) and backend services, the platform enables seamless communication between users, services, datasets, AI models, and deployment infrastructures. The updated Requirements Traceability Matrix (RTM) and the architectural validation activities

presented in this deliverable demonstrate that the final platform satisfies functional (FR) and non-functional requirements (NFRs) identified during the project. As a result, RAIDO delivers a mature, modular, and operational ecosystem capable of supporting trustworthy, explainable, interoperable, and sustainable AI across heterogeneous cloud, edge, and hybrid computing environments. The resulting architecture establishes a unified foundation for the development, optimization, deployment, monitoring, and governance of AI solutions while illustrating how trustworthy and sustainable AI can be effectively operationalized across diverse industrial and scientific domains.

1 Introduction

The “Final RAIDO Platform and Architecture” deliverable (D) 2.4 presents the final architecture and implementation status of the RAIDO platform, developed within Task (T) 2.3, “Overall RAIDO Platform Architectural Design”, of Work Package (WP) 2. Building upon the architectural foundations established in D2.3, this deliverable documents the evolution of the platform from its initial architectural design to its fully integrated and validated implementation. It consolidates the technical developments, component integrations, deployment activities, and architectural refinements performed throughout the project, providing a comprehensive description of the final RAIDO platform and its underlying architecture.

RAIDO delivers a trustworthy, explainable, and energy-efficient Artificial Intelligence (AI) ecosystem capable of supporting diverse Industry 4.0 application domains, including smart energy grids, smart farming and fungal food production, healthcare and pharmacogenomics, and robotics-based fibre characterization. To achieve this objective, the project combines advanced data optimization techniques, trustworthy AI mechanisms, explainability services, orchestration capabilities, distributed computing infrastructures, and monitoring solutions into a unified and modular platform architecture.

This deliverable describes the final architecture as realized following the implementation, integration, deployment, and validation activities carried out across WPs 3, 4, 5, and 6. It presents the platform components, their internal operation, interactions, interfaces, deployment considerations and architecture, and integration status, while highlighting the improvements introduced since D2.3. Particular emphasis is placed on demonstrating how the various individual technological building blocks, developed throughout the project, operate together as a cohesive ecosystem supporting reliable, trustworthy, explainable, and sustainable AI lifecycle management.

1.1 Objectives of the Deliverable

The objective of this deliverable is to provide a comprehensive description of the final RAIDO platform architecture and its constituent components, reflecting the outcome of the design, implementation, integration, and validation activities performed during the project. Whereas D2.3 introduced the initial architectural design and specification of the platform, D2.4 documents the final architecture as implemented, integrated, and evaluated within the RAIDO ecosystem.

The deliverable presents the overall platform structure, the relationships between its different architectural layers, and the interactions among the platform services responsible for data management, AI optimization, synthetic data generation, federated and continual learning, explainability, benchmarking, orchestration, visualization,

resource management, and blockchain-enabled traceability. Furthermore, it captures the evolution of each component by highlighting the additions, enhancements, and refinements introduced during the project, thereby providing a complete, clear view of the platform's final operational capabilities.

Another objective is to demonstrate how the technological outcomes developed across the RAIDO work packages have been integrated into a unified architecture capable of supporting trustworthy and green (energy-efficient) AI workflows across cloud, edge, and hybrid infrastructures. In this context, the deliverable serves as the architectural reference document for the final RAIDO platform, documenting component interoperability, information exchange mechanisms, deployment considerations, and the overall system behavior supporting efficient AI model optimization, deployment, monitoring, and continuous improvement.

Finally, the deliverable establishes the traceability between the final platform architecture and the technical requirements identified during the early stages of the project, demonstrating that the implemented system satisfies functional, non-functional, operational, and sustainability objectives defined for the RAIDO framework.

Throughout this deliverable, the platform components are referred to using their final operational names adopted during platform integration and validation activities (cf. D6.1). Each architectural layer, its corresponding components, and their operational names are presented in Section 2.3. These include ENRICH (for the Data Enrichment, Curation, Distillation and Federated Mining), DATAGEN (for the Synthetic Data Generation), DISTIL (for the Image and Time-Series Distillation), FEDCL (for the Federated and Continual Learning), DATALAKE, XAI (explainable AI), RLFEED (for the Reinforced Benchmarking and Feedback-Based Progress Monitoring), UI (for the Adaptive AI Processes and Visualization Engine), ORCH (for the Self-Evolving Green-AI Orchestrator), RSCMANAGER (for the Resource Allocation and Deployment), and BLC (for the Blockchain and Smart Contracts). These operational names are used consistently throughout the remainder of the document to reflect the final integrated implementation of the platform. Table 1 summarizes the mapping between the architectural layers, operational component names, and their corresponding platform components.

Table 1: Mapping between RAIDO architectural layers, operational component names, and corresponding platform components.

Layer	Operational Component Name	Component
User Interaction Layer	UI	Adaptive AI Processes and Visualization Engine
Data Preparation & AI Optimization Layer	ENRICH	Data Enrichment, Curation, Distillation and Federated Mining
Data Preparation & AI Optimization Layer	DATAGEN	Synthetic Data Generation

Data Preparation & AI Optimization Layer	DISTIL	Few- & Zero-Shot Adaptation and Distillation of Vision-Language Models (Energy & Compute Efficient Learning)
Data Preparation & AI Optimization Layer	FEDCL	Life-Long Learning in Federated Architectures (Federated & Continual Learning)
Data Management Layer	DATALAKE	Data Lake
Trustworthy AI Layer	XAI	Explainable Artificial Intelligence (XAI)
Trustworthy AI Layer	RLFEED	Reinforced Benchmarking and Feedback Monitoring
Integration & Orchestration Layer	ORCH	Self-Evolving Green-AI Orchestrator
Deployment & Resource Management Layer	RSCMANAGER	Resource Allocation and Deployment

1.2 Relation with other tasks and deliverables

D2.4 constitutes the final outcome of T2.3, "Overall RAIDO Platform Architectural Design", which spans from Month (M) 3 to M30 of the project. While D2.3 established the initial architectural blueprint of the RAIDO platform, the present deliverable captures the final architecture following the completion of the implementation, integration, deployment, and validation activities.

The deliverable builds upon D2.1, "Report on the Specifications of Pilots, Specifications, KPIs, Requirements and Novel Enabling Technologies", which defined the user requirements, non-functional requirements (NFRs), pilot needs, performance indicators, and enabling technologies that guided the architectural design of the platform.

It is also closely related to D2.2, "RAIDO Green AI Framework and Optimised Pipeline", which introduced and established the conceptual foundations of the RAIDO optimization framework and its energy-efficient and trustworthy AI lifecycle that are reflected throughout the platform architecture.

The architectural components described in this deliverable incorporate the technical developments performed within WP3, "Data Enrichment Solutions and Optimised Trustworthy AI Architectures". In particular, D3.3, "Data Lakes for Data and Model Storage, Analytics and Security", defines the Data Lake infrastructure and backend services that form the information management layer of the RAIDO platform.

The deliverable further consolidates the outcomes of WP4, "Human-centred and Ethical AI". Specifically, D4.2, "Trustworthy XAI and Benchmarking Progress Monitoring", provides the explainability, transparency, benchmarking, and feedback-driven monitoring capabilities integrated into the final platform architecture.

Similarly, D2.4 incorporates the outputs of WP5, "Energy-efficient E2C Deployment and Green AI Orchestration". D5.1, "Design and Implementation of the RAIDO AI Framework and Platform", provides the foundation for the platform implementation and visualization framework; D5.2, "Hybrid Cloud-Edge for Green AI Orchestration and

Resource Management", contributes the orchestration, resource allocation, networking management, and deployment mechanisms; and D5.3, "Transparent Process Monitoring with Blockchain Smart Contracts and IPFS Distributed Networking", provides the blockchain-enabled traceability and monitoring infrastructure.

Finally, the deliverable is strongly connected with D6.1, "Report on Pilot Setup, Deployment, Execution and Evaluation with RAIDO KPIs", which documents the deployment, integration, testing, and validation of the platform components within the RAIDO pilot environments. The architecture presented in this deliverable reflects the final integrated system as validated through these pilot activities. While the present deliverable focuses on documenting the final architecture of the RAIDO platform, its technological components, and their integration into a unified operational ecosystem, the deployment methodology, pilot validation activities, and the assessment of the project Key Performance Indicators (KPIs) are presented in D6.1. The corresponding validation outcomes and KPI evaluations provide complementary evidence supporting the realization of the proposed architecture, whereas their continued monitoring and reporting are addressed through the relevant technical and pilot deliverables. Consequently, these aspects are not repeated in the present deliverable, which is dedicated to the architectural realization and evolution of the RAIDO platform.

1.3 Document structure

The remainder of this document is organized as follows.

Section 2 presents the final RAIDO platform architecture, including the architectural layers, reference architecture, actors, component interactions, communication mechanisms, and end-to-end operational workflow.

Section 3 provides a detailed description of each platform component, including its purpose, architectural principles that guided its design, interactions with other components, and the additions and improvements introduced throughout the project.

Section 4 presents the final platform deployment and integration architecture, discussing deployment considerations, integration status, security mechanisms, scalability, and extensibility aspects.

Section 5 presents the updated Requirements Traceability Matrix (RTM) together with the architectural validation activities, demonstrating the alignment of the final platform with the functional (FRs) and non-functional requirements (NFRs) established during the project.

Finally, Section 6 summarizes the main architectural achievements, discussing the evolution of the platform since D2.3, and outlining future directions for the RAIDO ecosystem.

2 Architecture componentization

2.1 RAIDO platform overview

The RAIDO platform has been designed as a modular and service-oriented framework for Reliable AI and Data Optimization, integrating advanced data management, trustworthy AI, AI optimization, orchestration, deployment, monitoring, and governance capabilities within a unified ecosystem. The platform supports the complete AI lifecycle, enabling users to manage datasets, develop and optimize AI models, deploy them across distributed infrastructures, monitor their execution, and continuously improve their performance through integrated feedback and governance mechanisms.

Unlike conventional AI platforms that primarily address isolated stages of AI development, RAIDO adopts a holistic architectural approach that combines data-centric and model-centric optimization within a unified ecosystem. The platform supports the complete lifecycle of datasets and AI models through interoperable services responsible for data preparation, optimization, deployment, monitoring, and governance, while enabling seamless execution across cloud, edge, and hybrid infrastructures.

The architecture has been designed to support multiple Industry 4.0 application domains that generate highly heterogeneous datasets and impose different operational, computational, and regulatory requirements, including smart energy grids, smart farming and fungal food production, healthcare and pharmacogenomics, and robotics-based fibre characterization. To accommodate these diverse scenarios, the platform supports a wide range of data modalities, AI workloads, and deployment models spanning cloud, edge, and hybrid computing environments while maintaining a common architectural foundation.

At the architectural level, the RAIDO platform is organized into a set of specialized components, each responsible for a distinct aspect of the AI lifecycle, including data preparation, AI optimization, orchestration, deployment, trustworthiness, governance, and user interaction. Collectively, these components form a modular and layered architecture that supports the execution of integrated AI workflows across heterogeneous application domains. The architecture is centered around three complementary pillars: the DATALAKE, which provides centralized information management; the Backend Services and API Layer, which enables secure and standardized communication between platform services; and the orchestration and deployment infrastructure, which coordinates workflow execution and resource management. Together, these architectural elements enable the specialized platform components to operate as a unified ecosystem while supporting interoperability, scalability, and distributed execution.

The final architecture presented in this deliverable reflects the evolution of the initial architectural blueprint introduced in D2.3 into a fully integrated and operational platform validated through the RAIDO pilot activities. The following sections describe the final reference architecture, its architectural layers, component interactions, and operational workflows that collectively enable reliable, trustworthy, explainable, and sustainable AI across distributed computing environments.

2.2 Final RAIDO platform architecture

The initial RAIDO architecture, presented in D2.3, introduced the conceptual organization of the platform along with the internal architecture of its technical components and the information flows supporting data and AI optimization processes. Building upon this foundation (cf. Figure 1), the final architecture presented in Figure 2 reflects the outcome of the implementation, integration, and validation activities performed throughout the project. It consolidates the individual technological developments into a unified, layered, service-oriented platform and illustrates the operational relationships between the user(s), platform technological components and services, backend infrastructure, and deployment environments.

Compared with the initial architectural design, the final reference architecture introduces a more mature and modular organization of the platform by grouping the components into logical architectural layers with clearly defined responsibilities. This layered organization improves interoperability, maintainability, and scalability while allowing the individual components to evolve independently and participate in coordinated end-to-end data and AI workflows. A dedicated Backend Services and API Layer further strengthens this modular design by providing standardized communication interfaces that decouple platform services from the underlying infrastructure and enable consistent information exchange across the ecosystem.

At the core of the architecture lies the DATALAKE, which serves as the central information management layer of the platform. Rather than acting solely as a storage repository, the DATALAKE provides a unified environment for managing datasets, AI models, metadata, optimization artefacts, monitoring information, benchmarking results, and deployment resources generated throughout the AI lifecycle. The platform adopts a polyglot persistence approach by integrating MySQL, MongoDB, MinIO, and InfluxDB, while Keycloak provides centralized authentication and authorization services. Access to these resources is mediated through the Backend Services and API Layer, implemented using FastAPI, which exposes standardized Representational State Transfer (REST) Application Programming Interfaces (APIs) supporting dataset and model management, synthetic data generation, distillation, fairness assessment, monitoring, reporting, and user management. This service layer establishes a common communication mechanism through which all platform components exchange information without requiring direct point-to-point integration.

Building upon this information management backbone, the platform incorporates a collection of specialized services supporting the different stages of the data and AI lifecycle. ENRICH, DATAGEN, DISTIL, and FEDCL provide complementary capabilities for data preparation, synthetic data generation, model adaptation, and federated and continual learning, respectively. Specifically, ENRICH provides data enrichment, curation, distillation, and federated mining capabilities designed to improve dataset quality and reduce redundancies. DATAGEN supports the creation of synthetic data through generative models and digital twin (DT) technologies, enabling data augmentation and addressing data scarcity challenges. DISTIL provides efficient model adaptation and optimization techniques, while FEDCL supports federated and continual learning workflows that enable distributed and privacy-preserving AI model evolution.

Their execution is coordinated by ORCH, which orchestrates optimization workflows, manages dependencies between services, schedules component execution, and supervises information exchange across the platform. This orchestration mechanism, operating under the Green-AI paradigm shown in Figure 2, enables individual services to operate independently while participating in integrated optimization pipelines tailored to the requirements of each application scenario.

The architecture further incorporates a dedicated trustworthiness and governance domain through the XAI, RLFEED, and BLC components. XAI provides explainability and interpretability capabilities, allowing users to understand model decisions and evaluate system behavior. RLFEED complements these capabilities by continuously assessing optimization outcomes, benchmarking model performance, and supporting feedback-driven improvement processes. BLC provides blockchain-based monitoring, provenance management, and traceability services through immutable record keeping and transparent tracking mechanisms. Their integration within the common architectural framework ensures that transparency, accountability, reproducibility, and governance are embedded throughout the AI lifecycle rather than being treated as standalone post-processing activities.

Deployment and execution across heterogeneous computing infrastructures are supported and handled by the RSCMANAGER, which coordinates resource allocation, workload scheduling, and infrastructure utilization within Kubernetes-based cloud, edge, and hybrid environments. Together with Prometheus-based monitoring and Kepler, this layer enables scalable execution of AI workloads while supporting efficient resource utilization and operational resilience.

The primary entry point to the platform is provided by the Visualization Engine (UI), implemented as a Vue.js-based web application. The UI provides a unified interaction environment through which users access, configure, and supervise the platform services according to their assigned roles and permissions. By abstracting the complexity of the underlying infrastructure and relying on the Backend Services and API Layer for

communication, it enables intuitive interaction with the platform without exposing the implementation details of the underlying components.

Overall, the final RAIDO architecture represents the transition from the conceptual blueprint introduced in D2.3 to a fully integrated and operational platform architecture validated through the project pilots. Its modular, service-oriented, and layered design enables the coordinated execution of reliable, trustworthy, explainable, and sustainable AI workflows while providing the flexibility required to support heterogeneous application domains and distributed computing environments.

2.3 Architectural layers

To improve modularity, scalability, interoperability, and maintainability, the final RAIDO platform adopts a layered architectural model that separates responsibilities across distinct functional domains. Rather than organizing the platform around individual technologies, the architecture groups related services according to their role within the AI lifecycle. This separation of concerns enables independent component evolution, facilitates integration across diverse technologies, and simplifies the deployment in distributed cloud, edge, and hybrid computing environments. Each architectural layer provides a well-defined set of services while interacting with adjacent layers through standardized APIs and orchestration mechanisms. Together, these layers establish a cohesive framework supporting data management, AI optimization, trustworthiness, deployment, and user interaction throughout the complete lifecycle of AI applications.

2.3.1 User interaction layer

The User Interaction Layer constitutes the primary interface between users and the RAIDO platform. Rather than interacting directly with individual backend services, users access the platform through the Adaptive AI Processes and Visualization Engine (hereafter UI), which provides a unified environment for configuring workflows, managing datasets and AI models, monitoring execution, and analyzing platform outputs. This layer abstracts the complexity of the underlying infrastructure by exposing platform functionality through intuitive visualization and management interfaces. In addition to supporting operational activities, it enables Human-in-the-Loop (HITL) interactions, allowing users to review explainability results, assess benchmarking outcomes, validate optimization decisions, and provide feedback that contributes to the continuous improvement of AI workflows.

By decoupling user interactions from backend services, the User Interaction Layer improves usability while maintaining a clear separation between presentation, business logic, and infrastructure management.

2.3.2 AI optimization layer

The AI Optimization Layer constitutes the computational core of the RAIDO platform, providing the services responsible for preparing data and optimizing AI models prior to deployment. It includes the ENRICH, DATAGEN, DISTIL, and FEDCL components, which collectively support data enrichment and curation, synthetic data generation, model adaptation, knowledge distillation, federated learning, and continual learning. Architecturally, this layer transforms raw datasets into optimized AI assets through coordinated data-centric and model-centric optimization processes. The individual services operate independently but are orchestrated as part of integrated workflows, enabling flexible execution according to the requirements of different application domains.

The outputs generated by this layer, including enriched datasets, synthetic data, optimized models, and associated metadata, are subsequently consumed by the deployment, monitoring, and governance services operating in the upper architectural layers.

2.3.3 Trustworthiness and governance layer

The Trustworthiness and Governance Layer ensures that AI workflows remain transparent, accountable, explainable, and continuously evaluated throughout their operational lifecycle. The layer comprises the XAI, RLFEED, and BLC components, which complement the optimization process by providing explainability, benchmarking, performance assessment, provenance management, and blockchain-enabled traceability.

Unlike traditional AI platforms where trustworthiness is often addressed after model development, RAIDO integrates these capabilities directly into the platform architecture. As optimization workflows are executed, this layer continuously evaluates model behavior, records operational events, and generates the evidence required for auditing, reproducibility, and regulatory compliance.

Through its close integration with the optimization and deployment layers, the Trustworthiness and Governance Layer embeds transparency and accountability across the complete AI lifecycle rather than treating them as isolated evaluation activities.

2.3.4 Data management layer

The Data Management Layer provides the information backbone of the RAIDO platform through the DATALAKE component. It is responsible for storing, organizing, managing, securing, and providing controlled access to the datasets, AI models, metadata, monitoring information, benchmarking results, deployment artefacts, and optimization outputs generated during platform operation.

Rather than functioning solely as a storage repository, the DATALAKE enables information exchange between platform services while supporting governance, versioning, lifecycle management, analytics, and security. By integrating multiple complementary storage technologies within a unified management framework, the platform accommodates heterogeneous data modalities and the diverse requirements of its application domains.

As a shared information layer, the DATALAKE enables all architectural components to exchange information through common backend services and APIs, eliminating unnecessary point-to-point communication and improving the scalability and maintainability of the overall platform.

2.3.5 Integration and orchestration layer

The Integration and Orchestration Layer coordinates the execution of platform services and manages the interactions between the architectural layers. Its central component, ORCH, orchestrates end-to-end AI workflows by scheduling component execution, managing dependencies, coordinating information exchange, and supervising workflow progression. Rather than embedding workflow logic within individual services, RAIDO centralizes orchestration through this layer, allowing components to remain modular and loosely coupled while participating in complex optimization pipelines. Communication between services is achieved through standardized APIs and backend services, ensuring interoperability regardless of the underlying implementation technologies.

By coordinating both data flows and control flows, the Integration and Orchestration Layer provides the operational backbone that enables the different platform services to function as a single integrated ecosystem.

2.3.6 Deployment and infrastructure layer

The Deployment and Infrastructure Layer provides the computational resources required for executing AI workloads across cloud, edge, and hybrid computing environments. It is centered around the RSCMANAGER component, which is responsible for resource allocation, workload scheduling, infrastructure management, and deployment coordination. This layer abstracts the complexity of heterogeneous execution environments by dynamically allocating computational resources according to workload requirements while supporting efficient utilization of distributed infrastructures. Networking management services further ensure reliable communication between geographically distributed platform components and deployed AI services.

Together, these capabilities establish the execution environment upon which the remaining architectural layers operate, providing the scalability, resilience, and resource efficiency required for the deployment and continuous operation of the RAIDO platform.

2.3.7 Backend Services and API Layer

The Backend Services and API Layer constitutes the communication backbone of the RAIDO platform, providing standardized interfaces through which users, platform components, and external services interact with the underlying infrastructure. Rather than allowing direct communication between individual services, this layer centralizes business logic, request handling, authentication, authorization, and data access, thereby improving modularity, interoperability, maintainability, and scalability across the platform.

Implemented using the FastAPI framework, the layer exposes a comprehensive collection of REST APIs supporting the core functionality of the RAIDO ecosystem. These include APIs for user authentication and authorization, dataset and metadata management, AI model management, synthetic data generation, data distillation, fairness assessment, monitoring, reporting, workflow execution, and platform administration. By exposing these capabilities through standardized interfaces, the platform enables both the UI and the internal platform services to access common functionality independently of their underlying implementation technologies. The Backend Services and API Layer also mediates access to the DATALAKE, ensuring consistent interaction with the platform's polyglot persistence infrastructure while enforcing security policies through the integration of Keycloak-based identity and access management. This approach abstracts the complexity of the underlying storage technologies and provides a unified mechanism for managing datasets, AI models, metadata, monitoring information, benchmarking results, and other platform artefacts throughout the AI lifecycle.

Beyond user-facing interactions, the layer facilitates communication between the specialized platform components by exposing service endpoints that support information exchange throughout the platform. Components such as ENRICH, DATAGEN, DISTIL, FEDCL, XAI, RLFEED, ORCH, RSCMANAGER, and BLC interact with the backend services to retrieve and update platform resources, exchange metadata, trigger workflow execution, and access monitoring and reporting functionalities. Consequently, the Backend Services and API Layer establishes a loosely coupled, service-oriented architecture in which individual components remain autonomous while operating as part of a unified and interoperable ecosystem.

The backend architecture and the complete specification of the REST APIs exposed by the platform are described in detail in D3.3; it presents the design of the backend services, the API endpoints, their functionality, communication mechanisms, and the interaction between the platform components and the DATALAKE. Consequently, the present deliverable focuses on the architectural role of the Backend Services and API Layer within the overall RAIDO platform, while the detailed API specifications are provided in D3.3.

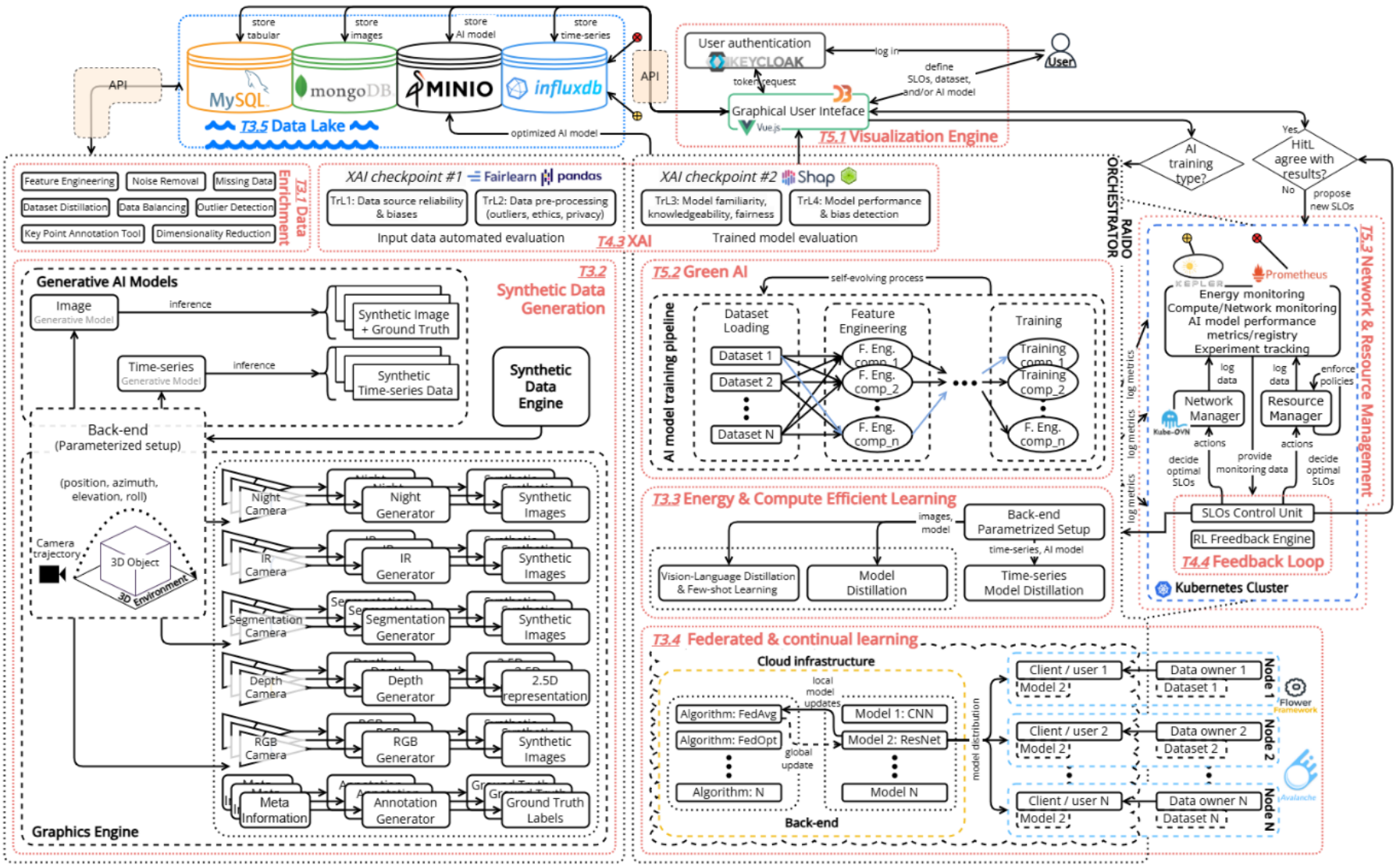

Figure 1: RAIDO initial reference architecture (v1).

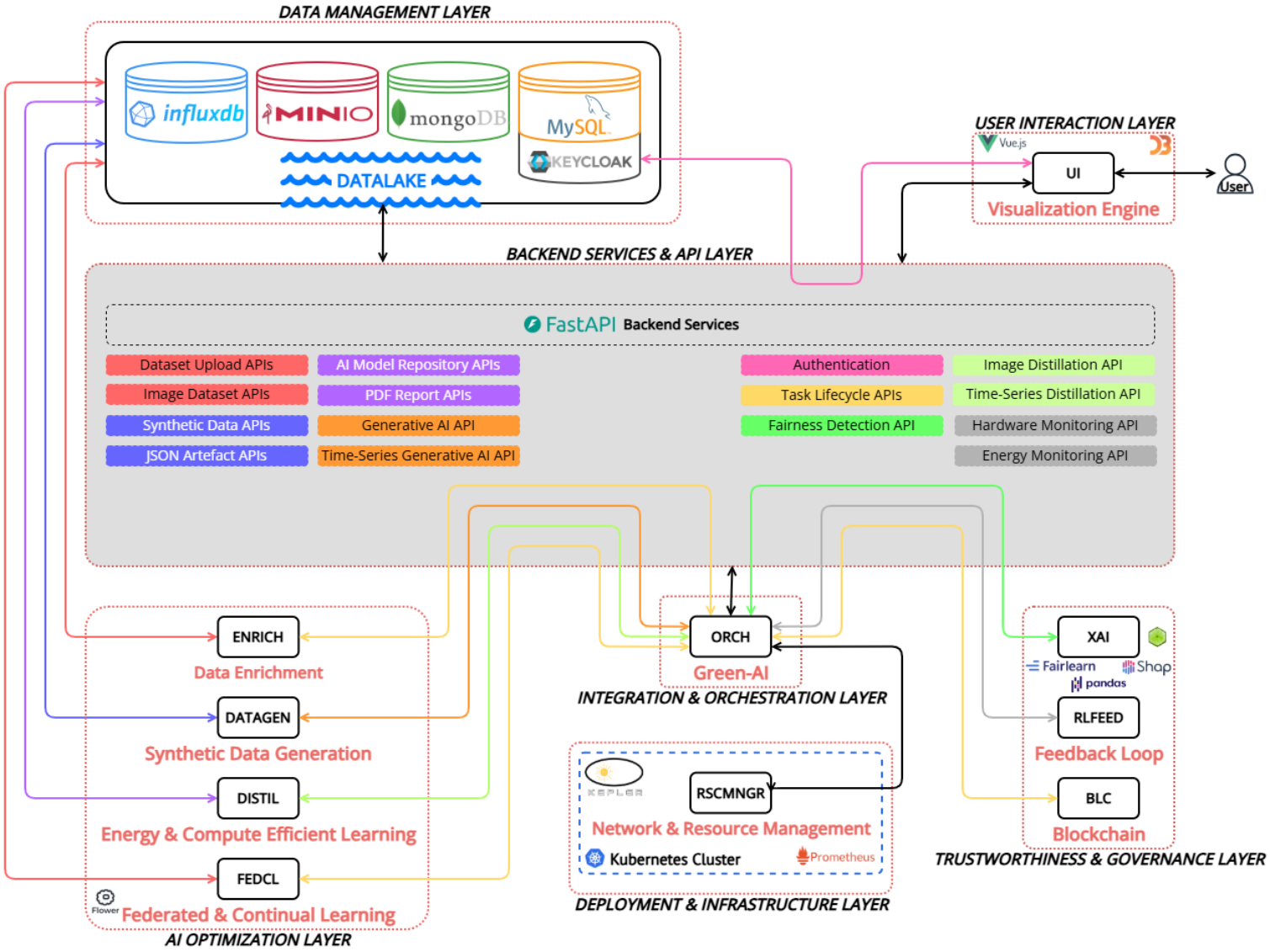

Figure 2: RAIDO final reference architecture (v2).

2.4 Architecture walkthrough & RAIDO actors

The layered architecture of the RAIDO platform enables different categories of users to access a common set of services while exposing only the functionality relevant to their responsibilities. Although the platform supports multiple application domains, all users interact with the platform through a common entry point, the Visualization Engine (UI), which provides unified access to the platform services according to the permissions associated with each user role.

A typical interaction begins when a user authenticates through the UI. Authentication and authorization are managed through the Backend Services and API Layer in conjunction with Keycloak, ensuring secure and role-based access to platform resources. Once authenticated, users can access the functionalities required for their respective activities, including data management, AI optimization, workflow execution, deployment supervision, explainability analysis, benchmarking, and platform administration. The Backend Services and API Layer transparently routes user requests to the appropriate platform services, while the complexity of the underlying platform architecture remains hidden from the user.

The technical actors and user roles of the RAIDO ecosystem were originally defined in D2.3 based on the operational requirements of the project pilots. These actor categories remain valid in the final platform implementation and continue to represent the primary stakeholders interacting with the system. The six (6) technical actors comprise Platform Administrators (PA), AI Engineers (AE), Data Engineers (DE), Data Providers (DP), Data Consumers (DC), and End Users.

Although all actors access the platform through the common user interface, their interactions with the underlying platform services differ according to their operational responsibilities.

- Platform Administrators (PA) oversee the deployment, configuration, security, and operational management of the platform, interacting primarily with the orchestration, deployment, monitoring, and administrative services.
- AI Engineers (AE) configure and execute AI optimization workflows, manage AI models, and evaluate optimization results using the platform's AI optimization and explainability services.
- Data Engineers (DE) are responsible for data preparation, enrichment, quality improvement, and lifecycle management, interacting mainly with the data optimization and data management services.
- Data Providers (DP) contribute datasets and associated metadata to the platform, ensuring that the information required by the AI workflows is available and properly maintained.

- Data Consumers (DC) analyze optimization results, benchmarking reports, and explainability outputs to support domain-specific decision making.
- End Users interact primarily with the visualization services, accessing the results produced by the platform according to their application-specific requirements without requiring knowledge of the underlying technical implementation.

The mapping between the technical actors defined in D2.3 and the principal platform services within the final architecture is summarized in Table 2. This mapping illustrates how the layered architecture provides role-based access to the platform while maintaining a unified interaction model across all user categories. Although the actors interact with different subsets of the platform services, they all follow the same architectural access path through the UI and the Backend Services and API Layer, ensuring consistent access control, interoperability, and usability throughout the platform.

Table 2: Mapping of D2.3 technical actors to the final RAIDO platform components.

D2.3 technical actor	Main interactions/components in final platform
Platform Administrator (PA)	UI, ORCH, DATALAKE, RSCMANAGER, BLC
AI Engineer (AE)	UI, ENRICH, DATAGEN, DISTIL, FEDCL, DATALAKE, XAI
Data Engineer (DE)	ENRICH, DATAGEN, DATALAKE, UI
Data Provider (DP)	UI, DATALAKE, ENRICH
Data Consumer (DC)	UI, XAI, RLFEED, DATALAKE
End User	UI, XAI, RLFEED, DATALAKE

2.5 Architectural integration principles

The final RAIDO platform adopts a modular, service-oriented architecture that enables independently developed technological components to operate as a unified ecosystem. Rather than relying on tightly coupled point-to-point communication, the platform integrates its services through standardized communication interfaces, shared information management, centralized orchestration, and distributed deployment mechanisms. This architectural approach enhances interoperability, maintainability, scalability, and extensibility while allowing individual components to evolve independently without affecting the overall platform.

Integration between the architectural layers is achieved through the Backend Services and API Layer, which provides a unified communication mechanism for both the UI and the specialized platform services. By exposing standardized REST APIs, the backend abstracts the implementation details of the individual components and provides secure, technology-independent communication across the platform. The backend architecture and the complete specification of the exposed REST APIs are presented in detail in D3.3, whereas the present deliverable focuses on their role within the overall platform architecture.

A key integration principle of the RAIDO platform is the separation of responsibilities across the architectural layers. Data management, AI optimization, trustworthiness assessment, orchestration, deployment, and user interaction are implemented as distinct functional domains that communicate through well-defined interfaces rather than direct dependencies. This separation of concerns facilitates the reuse of platform services, simplifies system maintenance, and enables the seamless incorporation of new functionalities as the platform evolves.

The architecture further promotes interoperability through a centralized information management strategy, whereby platform services access and manage datasets, AI models, metadata, monitoring information, benchmarking results, and other platform artefacts through the DATALAKE. This common information repository ensures data consistency across the platform while supporting governance, traceability, versioning, and lifecycle management of AI assets.

Workflow coordination and deployment management are performed independently of the individual platform services, allowing optimization pipelines to be dynamically configured and executed according to the requirements of each application scenario. This architectural approach enables flexible workflow composition while preserving the modularity and reusability of the participating components, regardless of the underlying computing infrastructure.

Overall, the integration principles adopted by the RAIDO platform establish a cohesive architectural framework that enables heterogeneous technologies to operate as a single coordinated ecosystem. The combination of standardized communication interfaces, shared information management, centralized orchestration, and distributed deployment provides the flexibility required to support multiple application domains while ensuring reliable, scalable, and sustainable platform operation.

2.6 End-to-End platform workflow

The RAIDO platform supports the complete lifecycle of AI applications through an integrated workflow that combines data management, AI optimization, trustworthiness assessment, deployment, monitoring, and continuous improvement within a unified architectural framework. By coordinating the interactions between the architectural layers and the specialized platform services, the platform enables users to efficiently develop, optimize, deploy, evaluate, and maintain AI solutions across heterogeneous computing environments.

The workflow is initiated when a user accesses the platform through the UI and authenticates using the platform's identity and access management services. Following successful authentication, users can configure AI workflows, manage datasets and AI models, monitor execution, and supervise platform operations according to their

assigned roles and permissions. Once a workflow has been configured, the request is processed through the Backend Services and API Layer and forwarded for execution.

Workflow execution is coordinated by the ORCH component, which determines the execution sequence of the participating platform services according to the configured optimization pipeline. ORCH manages workflow dependencies, schedules and orchestrates the execution of data and AI pipelines, coordinates interactions between platform components, and enables feedback-driven optimization without embedding workflow logic within the individual services.

The workflow typically begins with the preparation and optimization of the available datasets. Existing datasets are retrieved from the DATALAKE and processed by ENRICH, which performs data enrichment, curation, preprocessing, balancing, and federated mining operations to improve data quality and usability. When additional training data are required, DATAGEN complements this process by generating synthetic datasets through generative AI models and Digital Twin (DT) technologies, addressing data scarcity, privacy constraints, and class imbalance while improving the robustness of subsequent AI training processes.

Following data preparation, the optimized datasets are utilized for AI model optimization. DISTIL performs model compression techniques that reduce computational complexity while preserving predictive performance, enabling efficient deployment on resource-constrained environments. FEDCL subsequently supports federated and continual learning workflows, allowing AI models to be collaboratively trained, continuously updated, and incrementally improved without requiring centralized access to all training data.

Throughout the execution of the workflow, the DATALAKE persistently stores and manages datasets, AI models, metadata, monitoring information, benchmarking results, and optimization artefacts generated by the participating platform services. This enables consistent information sharing between the architectural layers while supporting governance, versioning, and lifecycle management of AI assets throughout the platform.

During workflow execution, XAI generates explainability information that enables users to interpret model predictions, while RLFEED evaluates optimization outcomes through benchmarking and feedback-driven performance assessment. In parallel, BLC continuously records workflow events, provenance information, and execution metadata, ensuring transparency, traceability, accountability, and auditability throughout the AI lifecycle.

Following validation, RSCMANAGER determines workload placement, allocates computational resources, and manages deployment, migration, and scaling of AI services across cloud, edge, and hybrid infrastructures. ORCH subsequently coordinates the corresponding deployment workflows and execution dependencies,

enabling efficient resource utilization and scalable operation across heterogeneous computing environments.

Finally, the execution results, including optimized datasets, AI models, explainability reports, benchmarking results, deployment information, monitoring metrics, and provenance records are presented to the user through the Visualization Engine. Based on these outputs, users may refine datasets, modify optimization parameters, initiate additional optimization cycles, or redeploy improved AI models, thereby establishing a continuous feedback loop that supports the iterative improvement of AI solutions throughout their operational lifecycle.

The end-to-end workflow demonstrates how the architectural layers and specialized platform components cooperate to deliver integrated AI services across the complete lifecycle of data and models. Through the coordinated operation of data management, orchestration, optimization, deployment, monitoring, explainability, and governance services, the platform enables reliable and scalable AI workflows while maintaining interoperability, traceability, and resource-efficient execution across heterogeneous computing environments.

3 RAIDO components

The RAIDO platform is composed of a collection of specialized technological components that collectively realize the functionality of the overall platform. Working together, these components support the complete data-driven and model-driven AI optimization process, including data preparation, synthetic data generation, model optimization, explainability, benchmarking, orchestration, deployment, resource management, governance, and user interaction. While Chapter 2 presented the platform from a system-level perspective, focusing on its layered architecture, integration principles, and operational workflows, the present chapter examines each component individually and describes its role within the final RAIDO ecosystem.

The initial architectural design of these components was introduced in Deliverable D2.3, where their objectives, expected functionality, and interactions were defined as part of the overall platform architecture. Throughout the project, these conceptual designs evolved into fully implemented and integrated platform services through the activities performed in Work Packages 3, 4, and 5. The resulting components constitute the operational building blocks of the final RAIDO platform and collectively realize the architectural vision presented in this deliverable.

For each component, this chapter presents its purpose within the RAIDO framework, summarizes the architectural principles established during the design phase, describes its final implementation and integration status, highlights its evolution since Deliverable D2.3, and explains its interactions with the remaining platform services. Rather than examining the architecture as a whole, the following sections focus on the individual technological building blocks that collectively enable the operation of the final RAIDO platform.

In this Section, we delve into each technical component of the RAIDO architecture, presenting them in alignment with the RAIDO workplan, i.e. we begin with “T2.2: Core Optimized AI-Pipelines” and proceed through each task in sequence (i.e. “T3.1 Data enrichment, curation and distillation, and Federated Mining”, “T3.2 Synthetic Data Generation”, “T3.3 Few- & Zero-Shot adaptation and distillation of Vision-Language Models”, “T3.4 Life-Long Learning in Federated Architectures”, “T3.5 Data Lake”, “T4.3 Frameworks for AI explainability”, “T4.4. Reinforced benchmarking and feedback-based progress monitoring”, “T5.1 Adaptive AI processes and visualization engine”, “T5.2 Self-evolving green-AI and data orchestration”, “T5.3 Networking management and AI-based E2C continuum resource allocation & deployment”), ultimately reaching “T5.4: Blockchain & Smart Contracts with IPFS Support”, covering the business logic of all the components, their functional description, internal architecture and information flows, and interactions with other components.

3.1 Core Optimized AI-Pipelines

3.1.1 Purpose and role within RAIDO

The Core Optimized AI-Pipelines constitute the foundational optimization framework of the RAIDO platform and define the end-to-end AI lifecycle adopted throughout the project. The component provides the conceptual and operational basis for executing data-centric and model-centric AI optimization workflows across distributed edge, cloud, and hybrid computing environments.

The business logic behind the AI-Pipelines is designed to optimize the balance between edge computing efficiency and centralized model intelligence while simultaneously addressing scalability, performance, fairness, transparency, explainability, and sustainability requirements. By leveraging edge nodes for localized data processing, validation, feature engineering, and model training, the framework reduces latency, minimizes unnecessary data movement, and improves resource utilization. At the same time, centralized processing enables model aggregation, global optimization, performance evaluation, and continuous monitoring of deployed AI systems.

The component incorporates support for advanced Machine Learning paradigms including data distillation, transfer learning, lifelong learning, and model optimization. Furthermore, Ethical AI principles are integrated throughout the pipeline through privacy-preserving mechanisms, secure data management procedures, and HITL supervision capabilities. Explainability mechanisms based on techniques such as SHAP, LIME, and Grad-CAM further enhance transparency and user trust by providing interpretable predictions and model behavior explanations.

As a result, the Core Optimized AI-Pipelines establish a unified framework through which the technological components developed within WP3, WP4, and WP5 operate together, supporting reliable, trustworthy, explainable, and sustainable AI workflows across the diverse pilot environments addressed by RAIDO.

3.1.2 Architecture and design principles

The initial architecture of the Core Optimized AI-Pipelines was introduced in D2.3 and was designed as a modular Machine Learning (ML) pipeline capable of supporting the complete AI lifecycle, from data acquisition and preparation to model deployment and continuous optimization.

The architecture was organized around two (2) principal processing domains: an Edge Processing Layer and a Centralized Processing Layer. This separation was introduced to balance computational efficiency, scalability, and resource utilization while enabling the deployment of AI solutions across heterogeneous infrastructures.

The Edge Processing Layer was responsible for local data processing and model development activities. The pipeline begins with data loading operations, where raw data collected across distributed edge nodes is ingested and prepared for downstream processing. Data distillation mechanisms may additionally be applied to reduce dataset complexity, compress information, or refine the collected data prior to further processing. Following data loading, data validation procedures are performed. These include exploratory data analysis, outlier detection, anomaly investigation, and data schema generation. These activities ensure data consistency, improve overall data quality, and facilitate interoperability across multiple data sources and operational environments. The validated datasets subsequently enter the data transformation stage, where feature preprocessing and feature engineering operations are performed. Data cleaning, normalization, encoding, transformation, and feature extraction techniques are applied to generate optimized representations suitable for Machine Learning applications. Additional feature engineering procedures may be employed to improve predictive performance and increase the effectiveness of downstream learning algorithms. The transformed data is then utilized during the model training stage. This stage includes feature selection, algorithm selection, hyperparameter optimization, and model training procedures. Through these activities, the pipeline supports the development of efficient and accurate Machine Learning models while minimizing computational complexity and reducing resource consumption.

The Centralized Processing Layer complements the distributed edge activities by providing model aggregation, evaluation, optimization, and deployment capabilities. Models trained across individual edge nodes are aggregated into unified global models, enabling the consolidation of distributed knowledge and supporting collaborative learning processes. Performance metrics are calculated to evaluate model accuracy, robustness, efficiency, and generalization capability, while dedicated bias investigation mechanisms are employed to identify and mitigate potential fairness issues. Following evaluation, optimized models are updated and deployed through the inference pipeline. Model serving mechanisms allow the resulting AI models to be executed either within centralized environments or distributed edge infrastructures, depending on the operational requirements of the target application.

The architecture additionally incorporates two (2) cross-functional domains that operate throughout the entire optimization lifecycle. The Ethical AI domain introduces privacy-preserving mechanisms, secure data storage capabilities, and Human-in-the-Loop supervision processes that ensure compliance with trustworthy AI principles. The Explainability domain provides interpretable AI capabilities through techniques such as SHAP, LIME, and Grad-CAM, enabling users to understand model decisions and assess the reliability of generated predictions.

Collectively, these architectural elements establish a scalable, distributed, explainable, and trustworthy optimization framework capable of supporting heterogeneous AI workloads across multiple deployment environments.

The high-level architecture of this component is shown in Figure 3.

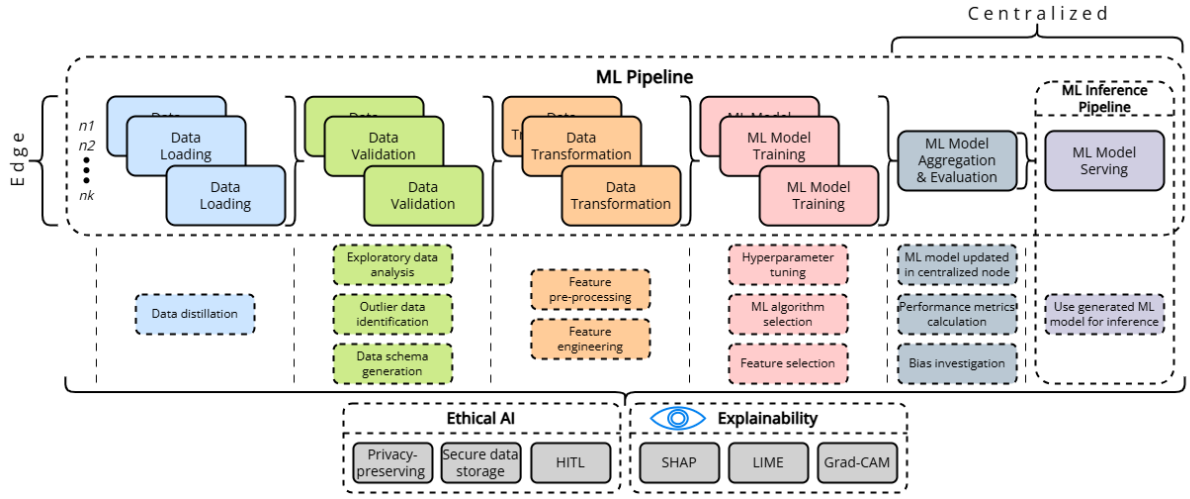

Figure 3: AI-pipelines component architecture.

3.1.3 Interactions with other components

The Core Optimized AI-Pipelines interact with a large number of services within the RAIDO ecosystem and serve as the conceptual framework through which the optimization activities of the platform are executed.

A primary interaction exists with the DATALAKE, which provides storage, retrieval, and lifecycle management services for datasets, trained models, metadata, benchmarking information, monitoring records, and optimization artefacts generated throughout the pipeline. The DATALAKE therefore acts as the central information repository supporting all stages of the optimization workflow.

The pipeline additionally interacts with the ENRICH component, which provides data enrichment, curation, distillation, and federated mining capabilities that improve the quality and usability of datasets prior to model training. Synthetic data generation and augmentation capabilities are provided through DATAGEN, enabling the pipeline to address data scarcity, imbalance, and privacy-related challenges.

Model-centric optimization activities are further supported through DISTIL and FEDCL. DISTIL contributes model adaptation, knowledge distillation, transfer learning, and model compression capabilities, while FEDCL enables federated and continual learning workflows that facilitate distributed training and continuous model evolution across heterogeneous environments.

The optimization process further interacts with XAI and RLFEED, which provide explainability, benchmarking, performance assessment, and feedback-driven optimization capabilities. These services enable the continuous evaluation of model behavior and support the realization of trustworthy AI objectives.

Workflow execution and coordination activities are performed through ORCH, which manages dependencies between services, schedules optimization tasks, and orchestrates the execution of end-to-end workflows. Deployment-related interactions are supported through RSCMANAGER and Networking Management services, enabling model deployment across cloud, edge, and hybrid infrastructures. Finally, BLC contributes provenance management, governance, monitoring, and traceability capabilities throughout the AI lifecycle.

Collectively, these interactions demonstrate how the Core Optimized AI-Pipelines provide the conceptual foundation upon which the remaining RAIDO platform services cooperate to execute integrated, trustworthy, and sustainable AI optimization workflows.

3.1.4 Additions, improvements and final version of the component

No significant architectural modifications were introduced to the Core Optimized AI-Pipelines component following the publication of D2.3. The original architecture successfully captured the requirements of the RAIDO optimization framework and therefore remained applicable throughout the implementation, integration, and validation phases of the project.

The architectural principles underpinning the component, including the separation between Edge Processing and Centralized Processing, the incorporation of data validation and transformation mechanisms, the support for distributed model training and aggregation, and the integration of Ethical AI and Explainability capabilities, were preserved throughout the project lifecycle. Likewise, the component continued to support advanced optimization paradigms including data distillation, transfer learning, lifelong learning, few-shot learning, bias investigation, performance monitoring, and model serving operations. Consequently, the original workflow structure required no major architectural redesign.

While the conceptual architecture remained stable, the supporting platform services matured significantly through the implementation activities performed within WP3, WP4, and WP5. Consequently, the functionality initially envisioned within the Core Optimized AI-Pipelines architecture is now realized through the coordinated operation of ENRICH, DATAGEN, DISTIL, FEDCL, XAI, RLFEED, DATALAKE, ORCH, RSCMANAGER, Networking Management, and BLC within the final RAIDO platform.

The successful implementation, integration, deployment, and validation of these services further confirmed the validity of the original design decisions and demonstrated

the suitability of the proposed optimization framework for supporting reliable, trustworthy, explainable, and sustainable AI workflows across heterogeneous application domains and computing environments.

3.2 Data Enrichment, Curation, Distillation, and Federated Mining

3.2.1 Purpose and role within RAIDO

The Data Enrichment, Curation, Distillation, and Federated Mining component, referred to as ENRICH in the final RAIDO platform, is responsible for optimizing the quality, usability, compactness, and representativeness of datasets utilized throughout the AI lifecycle. As AI systems are highly dependent on the quality of the underlying data, the component serves as one of the primary data-centric optimization services within the RAIDO ecosystem, ensuring that datasets are properly prepared before being consumed by downstream AI optimization and training processes.

The component addresses challenges commonly encountered in real-world datasets, including missing values, noise, outliers, redundancies, class imbalance, and data heterogeneity. Through a combination of enrichment, curation, transformation, distillation, and federated mining techniques, ENRICH improves the quality of available data while simultaneously reducing unnecessary complexity and storage requirements. This contributes directly to the development of more efficient AI models while reducing computational and energy consumption throughout the training and deployment phases.

The architecture of the component is organized around two complementary functional domains. The first domain focuses on Data Enrichment, Curation, and Distillation, aiming to improve data quality, increase information value, and create compact dataset representations suitable for AI model optimization. The second domain focuses on Federated Data Mining, enabling privacy-preserving discovery of relevant information and data resources across distributed environments. Through this functionality, the component can identify useful external data sources capable of addressing dataset deficiencies and imbalances while respecting privacy constraints and data ownership requirements.

As a result, ENRICH constitutes a key building block of the RAIDO optimization ecosystem and provides the high-quality data foundation required for subsequent model optimization, synthetic data generation, federated learning, explainability, benchmarking, and deployment activities.

3.2.2 Architecture and design principles

The initial architecture of the Data Enrichment, Curation, Distillation, and Federated Mining component was introduced in D2.3 and was designed as a modular preprocessing framework capable of supporting multiple data optimization workflows.

The architecture was intentionally designed to operate independently of specific AI models, allowing the generated outputs to be reused across different optimization pipelines and application domains.

The component was structured around two hierarchically equivalent subcomponents: the Dataset Enrichment, Curation, and Distillation module and the Federated Data Mining module. The execution order of these modules was not predetermined, allowing the RAIDO Orchestrator to dynamically activate the appropriate functionality according to the requirements of the optimization workflow being executed.

The Dataset Enrichment, Curation, and Distillation module was designed to improve both data quality and data compactness through a collection of preprocessing operations. These functionalities include missing data handling, noise removal, outlier detection, feature engineering, data transformation, and dataset distillation techniques. Dataset distillation is particularly important within the RAIDO framework, as it enables the generation of compact yet representative subsets of data that preserve critical information while reducing storage requirements and computational costs.

The Federated Data Mining module was introduced to address challenges associated with data scarcity and dataset imbalance while preserving privacy and data ownership constraints. Instead of directly accessing external datasets, the module was designed to discover relevant information through privacy-preserving mechanisms and metadata exchange procedures. This approach enables the identification of suitable data sources capable of addressing deficiencies within the local dataset without requiring direct access to sensitive information.

An important design principle of the component was its close interaction with the Data Lake and the Orchestrator. The component retrieves datasets from the Data Lake, performs the requested optimization operations, and subsequently stores the transformed datasets back into the platform repository. Metadata describing the executed preprocessing pipeline, applied transformations, and generated outputs is returned to the Orchestrator, enabling the orchestration layer to coordinate subsequent optimization steps within the broader AI lifecycle.

The architecture was designed to support multiple data modalities, including images, time-series data, tabular data, and textual information, ensuring applicability across the heterogeneous pilot environments addressed by RAIDO.

The high-level architecture of this component is shown in Figure 4.

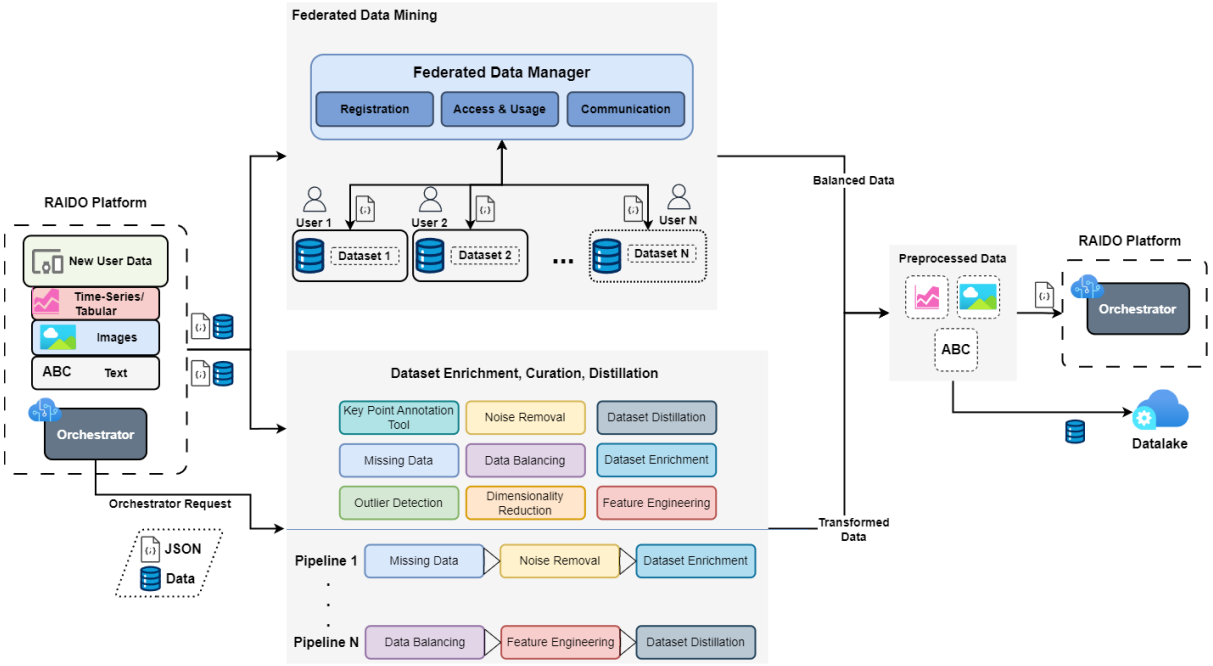

Figure 4: Data enrichment component architecture.

3.2.3 Interactions with other components

ENRICH interacts extensively with multiple platform services and plays a central role within the data-centric optimization ecosystem of RAIDO.

Its primary interaction occurs with ORCH, which acts as the central coordination mechanism of the platform. ORCH is responsible for triggering the appropriate enrichment, curation, distillation, and federated mining workflows according to the requirements of the optimization pipeline. Following the execution of the requested operations, ENRICH returns metadata describing the completed tasks, enabling ORCH to coordinate subsequent stages of the workflow and maintain awareness of the optimization process.

The component also interacts directly with the DATALAKE, which serves as the primary repository for datasets, metadata, and optimization artefacts. Prior to execution, ENRICH retrieves the required datasets from the DATALAKE and performs the requested preprocessing activities. Upon completion, the transformed datasets and generated artefacts are stored back into the DATALAKE, making them available for subsequent optimization stages and future reuse.

Close interactions additionally exist with DATAGEN, particularly in scenarios involving dataset imbalance or data scarcity. When data quality issues, imbalance, or insufficient training samples are identified during preprocessing, ENRICH stores the corresponding datasets and metadata in the DATALAKE. ORCH may subsequently invoke DATAGEN to generate synthetic data that address these deficiencies before continuing the optimization workflow.

The outputs produced by ENRICH are further utilized by DISTIL, FEDCL, and other optimization services as inputs for model training, adaptation, distillation, federated learning, and continual learning activities. Consequently, the component serves as one of the primary entry points to the optimization lifecycle and significantly influences the effectiveness of the downstream AI processes.

Through these interactions, ENRICH contributes to the creation of high-quality, representative, and computationally efficient datasets capable of supporting reliable and trustworthy AI development throughout the platform.

3.2.4 Additions, improvements, and final version of the component

The overall architecture and design principles of the Data Enrichment, Curation, Distillation, and Federated Mining component remained largely unchanged following the publication of D2.3. The original design successfully addressed the requirements associated with data quality improvement, dataset optimization, privacy-preserving data enhancement, and data-centric AI optimization, and therefore did not require significant architectural modifications during the subsequent development phases.

The primary evolution of the component occurred through its implementation, integration, and validation within the operational RAIDO platform. The conceptual architecture originally proposed in D2.3 was transformed into a fully functional service, referred to in the final platform as ENRICH, capable of executing enrichment, curation, distillation, and federated mining workflows across heterogeneous datasets including images, tabular data, time-series information, and textual content.

The Dataset Enrichment, Curation, and Distillation functionalities were implemented and integrated into the platform, enabling automated data quality improvement through mechanisms such as missing value handling, noise reduction, outlier detection, feature engineering, data transformation, and dataset distillation. These capabilities allow ENRICH to generate compact, representative, and high-quality datasets suitable for downstream AI optimization activities while reducing computational complexity and storage requirements.

Similarly, the Federated Data Mining functionality was implemented to support privacy-preserving discovery of relevant information across distributed environments. Through metadata exchange and federated analysis techniques, the component enables the identification of suitable external data sources without requiring direct access to sensitive datasets. When appropriate external data is unavailable, the generated metadata can be utilized by the Synthetic Data Generation service to facilitate the creation of representative synthetic samples capable of addressing identified dataset deficiencies.

A significant improvement compared to the initial design phase was the full integration of ENRICH within the broader RAIDO ecosystem. The component now operates in

conjunction with ORCH, DATALAKE, DATAGEN, DISTIL, FEDCL, XAI, RLFEED, and the remaining platform services, allowing its functionality to be incorporated into end-to-end optimization workflows spanning the complete AI lifecycle.

Consequently, while the underlying architecture remained largely unchanged, the component evolved from a conceptual design into a mature and operational platform service. The successful implementation, integration, deployment, and validation activities performed throughout the project confirmed the effectiveness of the original design decisions and established ENRICH as one of the core data optimization services within the final RAIDO platform architecture.

3.3 Synthetic Data Generation

3.3.1 Purpose and role within RAIDO

The Synthetic Data Generation component, referred to as DATAGEN in the final RAIDO platform, is responsible for generating high-quality synthetic datasets that complement and enhance the real-world data available within the RAIDO ecosystem. The component addresses one of the most common challenges encountered during AI development, namely the limited availability, imbalance, incompleteness, or privacy sensitivity of real-world datasets. By generating realistic and richly annotated synthetic data, DATAGEN supports the creation of more robust, reliable, and generalizable AI models while reducing the dependency on large volumes of manually collected data.

The business logic of the component is centered around the intelligent generation of synthetic data tailored to the specific needs of downstream AI workflows. Operating under the coordination of the RAIDO Orchestrator, the component dynamically selects the most appropriate generation mechanism according to the requirements of the requested task. Depending on the characteristics of the data and the desired outputs, synthetic data generation can be performed through either simulation-based approaches using digital twins and graphics engines or through advanced Generative Artificial Intelligence techniques.

To achieve this objective, DATAGEN combines two complementary generation paradigms. The first is a Graphics Engine capable of producing photorealistic synthetic environments with precise control over scene parameters such as object placement, camera trajectories, environmental conditions, lighting, and sensor configurations. The second is a Generative AI subsystem that utilizes advanced deep learning architectures, including Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Diffusion Models, to generate realistic synthetic images and time-series data that closely resemble real-world distributions.

Through the combination of these approaches, DATAGEN enables the creation of scalable, diverse, and highly annotated datasets capable of supporting a wide variety of

AI applications across the RAIDO pilots. The generated data contributes directly to dataset balancing, data augmentation, model robustness, and the overall effectiveness of the platform's optimization ecosystem.

3.3.2 Architecture and design principles

The initial architecture of the Synthetic Data Generation component was introduced in Deliverable D2.3 and was designed as a modular synthetic data generation framework capable of supporting multiple data modalities and generation strategies. The architecture was intentionally designed to provide flexibility in addressing different data generation requirements while ensuring seamless integration with the broader RAIDO optimization ecosystem.

At the center of the architecture lies the Synthetic Data Engine, which acts as the coordination layer responsible for receiving generation requests from the RAIDO Orchestrator and directing them to the most suitable generation subsystem. This design enables the component to dynamically select between simulation-based and AI-driven generation methodologies according to the specific characteristics of the requested task.

One of the primary architectural pillars of the component is the Graphics Engine. The Graphics Engine operates as a parameterized simulation environment capable of replicating complex real-world scenarios through the rendering of synthetic 3D environments. The architecture allows precise manipulation of simulation parameters including object positioning, environmental conditions, camera trajectories, viewing angles, illumination settings, and sensor characteristics. Through this capability, the Graphics Engine can generate a variety of outputs including RGB images, segmentation masks, depth maps, infrared imagery, nighttime imagery, and fully annotated datasets containing ground-truth labels. This approach is particularly valuable in situations where real-world data collection is expensive, impractical, unsafe, or impossible.

Complementing the Graphics Engine is the Generative AI Models subsystem. This subsystem provides a data-driven approach to synthetic data generation by leveraging advanced deep learning architectures capable of learning the underlying distributions of real-world datasets and reproducing them through synthetic samples. The architecture supports both fine-tuning of generative models using user-provided datasets and the utilization of pre-trained models optimized for specific tasks. Depending on the data modality and use-case requirements, different generative paradigms may be employed, including GANs, VAEs, and Diffusion Models.

For image generation tasks, the subsystem is capable of generating realistic synthetic images accompanied by annotations and ground-truth labels. For temporal applications, dedicated time-series generation models can produce synthetic sequences that preserve temporal dependencies and dynamic patterns observed in real-

world datasets. These capabilities allow the generation of large-scale synthetic datasets suitable for training, validation, benchmarking, and AI model optimization activities.

The overall architecture was designed to support scalability, flexibility, multimodality, and interoperability, ensuring that synthetic data generation can be incorporated seamlessly into the broader data-centric optimization workflows of the RAIDO platform.

The high-level architecture of this component is shown in Figure 5.

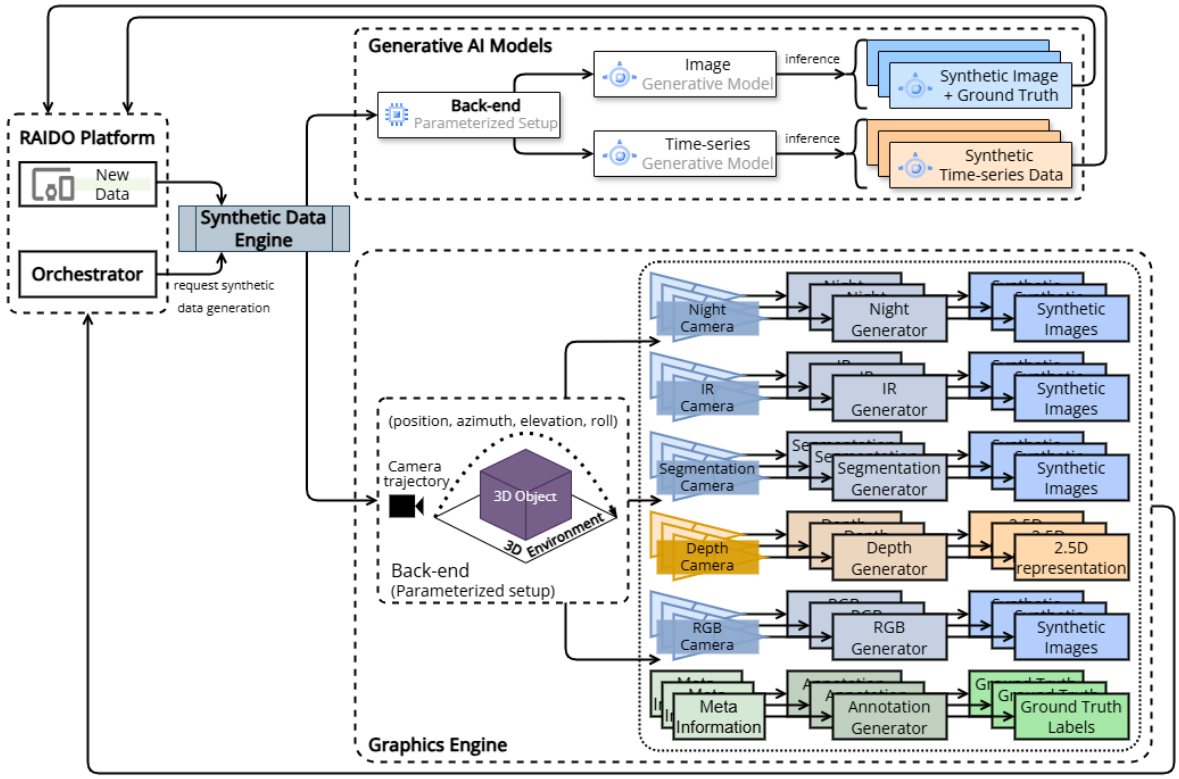

Figure 5: Synthetic data generation component architecture.

3.3.3 Interactions with other components

DATAGEN interacts closely with several services within the RAIDO ecosystem and plays a critical role in supporting data-centric optimization workflows.

Its primary interaction occurs with ORCH, which coordinates the execution of optimization workflows across the platform. The Orchestrator issues synthetic data generation requests whenever additional data is required to address challenges such as dataset imbalance, insufficient sample availability, lack of representative examples, or specific augmentation requirements. Based on the workflow configuration provided by ORCH, DATAGEN executes the appropriate synthetic data generation pipeline and returns the generated datasets for downstream processing.

The component additionally interacts with ENRICH, particularly in scenarios where dataset deficiencies are identified during data preprocessing and federated mining activities. Metadata generated by ENRICH regarding missing classes, underrepresented

categories, or data scarcity conditions can be utilized by DATAGEN to guide the synthetic data generation process and create representative synthetic samples tailored to the specific requirements of the dataset.

Generated datasets are stored and managed through the DATALAKE, which serves as the central repository for synthetic data, metadata, annotations, and generated artefacts. Through this interaction, synthetic datasets become available for reuse by other optimization services and future workflows executed within the platform.

The outputs produced by DATAGEN are subsequently consumed by downstream optimization services including DISTIL and FEDCL, where synthetic samples may be utilized for model training, transfer learning, continual learning, knowledge distillation, and federated learning activities. The generated datasets may also contribute to benchmarking, explainability, and evaluation processes performed by RLFEED and XAI.

Through these interactions, DATAGEN functions as a critical enabler of data augmentation and dataset optimization within the broader RAIDO ecosystem, supporting the creation of more representative, balanced, and robust datasets for AI development.

3.3.4 Additions, improvements, and final version of the component

The fundamental architecture and design principles of the Synthetic Data Generation component remained largely unchanged following the publication of Deliverable D2.3. The original design successfully captured the requirements associated with synthetic data generation, dataset augmentation, and data scarcity mitigation within the RAIDO ecosystem and therefore did not require significant architectural modifications during the subsequent implementation phases.

The primary evolution of the component occurred through its implementation, integration, and validation within the operational platform. The conceptual architecture originally described in D2.3 was transformed into a fully functional service, referred to as DATAGEN within the final RAIDO architecture. The implementation realized both the Graphics Engine and Generative AI generation pathways, enabling the generation of synthetic data across multiple modalities and supporting the requirements of the project pilots.

A significant advancement compared to the initial design phase was the integration of DATAGEN within the broader optimization ecosystem of RAIDO. The component now operates as part of coordinated end-to-end workflows involving ORCH, ENRICH, DATALAKE, DISTIL, FEDCL, XAI, RLFEED, and the remaining platform services. This integration allows synthetic data generation to be dynamically invoked whenever required as part of data optimization, dataset balancing, augmentation, model training, or evaluation processes.

The implementation and validation activities performed throughout the project demonstrated the ability of DATAGEN to generate realistic and highly representative synthetic datasets capable of supporting AI model development across multiple application domains. The generated outputs contribute directly to improved model robustness, increased dataset diversity, reduced dependence on manually collected data, and enhanced support for low-resource or imbalanced learning scenarios.

Consequently, while the underlying architectural principles remained consistent with those originally proposed in D2.3, the component evolved from a conceptual design into a mature and operational platform service. The successful implementation, integration, and validation of DATAGEN confirmed the effectiveness of the original design decisions and established the component as one of the key enablers of data-centric optimization within the final RAIDO platform.

3.4 Few- & Zero-Shot adaptation and distillation of Vision-Language Models

3.4.1 Purpose and role within RAIDO

The Few- & Zero-Shot Adaptation and Distillation of Vision-Language Models component, also referred to as Energy and Compute Efficient Learning and operationally named DISTIL within the final RAIDO platform, focuses on adapting AI models to downstream tasks with minimal data and energy consumption. The component forms part of the AI Model Optimisation Layer and supports efficient model training through knowledge distillation techniques and model adaptation mechanisms.

Integrated with the RAIDO Orchestrator (ORCH), DISTIL processes training data and AI models and routes them through appropriate distillation workflows to achieve energy-efficient model optimization. The component supports both vision and time-series applications, extending its applicability across a wide range of pilot scenarios. By reducing computational requirements while maintaining predictive performance, DISTIL contributes directly to the Green AI objectives of the RAIDO project.

The component leverages state-of-the-art technologies to achieve energy- and data-efficient training for both vision and time-series models. For vision applications, it utilizes Decoupled Knowledge Distillation (DKD), which separates logit distillation and feature distillation into independent optimization processes, enabling effective knowledge transfer from large teacher models to smaller and more efficient student models. For time-series forecasting applications, state-of-the-art architectures such as PatchTST, combined with distillation losses, are employed to improve both performance and efficiency.

Collectively, these technologies enable the AI Model Optimisation Layer to provide scalable, optimized, and energy-efficient training solutions while reducing the computational footprint of AI models deployed across the RAIDO pilot environments.

3.4.2 Architecture and design principles

The initial architecture of DISTIL was designed to support efficient adaptation and optimization of AI models through a combination of knowledge distillation, model compression, transfer learning, few-shot learning, and zero-shot learning techniques. The component was envisioned as a flexible optimization service capable of operating on both vision and time-series data while supporting different downstream AI tasks.

The architecture was integrated with the RAIDO orchestration framework, allowing optimization requests to be dynamically triggered according to user requirements and workflow configurations. Depending on the selected task, models and datasets could be routed through specialized distillation workflows designed to improve performance while reducing model size, training requirements, and energy consumption.

For vision models, the architecture adopted DKD as a core optimization strategy, enabling efficient transfer of knowledge from large teacher models to compact student models. For time-series forecasting tasks, PatchTST combined with distillation losses was selected as a suitable approach for generating efficient forecasting models capable of operating in resource-constrained environments.

The resulting design established the foundations for a flexible and energy-efficient model optimization framework aligned with the sustainability objectives of the RAIDO platform.

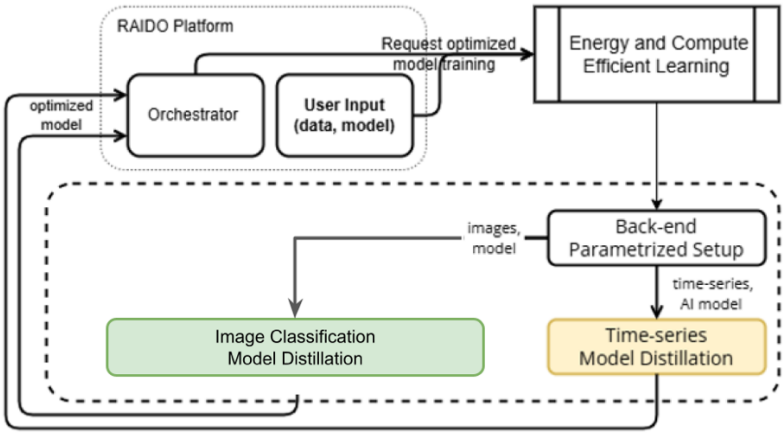

Figure 6: Energy and compute efficient learning architecture.

3.4.3 Interactions with other components

DISTIL operates under the coordination of ORCH, which invokes the component whenever model adaptation, optimization, or distillation activities are required. The Orchestrator forwards datasets, AI models, optimization parameters, and execution

configurations to the component and receives the resulting optimized models and execution outputs.

Through this interaction, DISTIL functions as a responsive optimization service within the RAIDO platform, supporting downstream workflows and enabling the efficient training and deployment of AI models across different application domains.

3.4.4 Additions, improvements, and final version of the component

Since D2.3, the component evolved towards a streamlined and resource-efficient optimization service aligned with the final integrated RAIDO platform architecture.

Prompt learning and cross-modal distillation were explored as code-level tools but were not included in the final integrated RAIDO platform workflow. Both approaches rely on CLIP-like Vision-Language Models and therefore require Vision-Language training or adaptation. Prompt learning depends on a Vision-Language backbone and optimizes learnable prompts through the text encoder, while cross-modal distillation requires both teacher and student Vision-Language Models and the preservation of visual-textual embedding alignment during training.

This refinement is aligned with the final RAIDO platform infrastructure and integrated workflow design, which prioritize resource-efficient and stable execution of platform components. Compared with the final integrated single-modality distillation workflows for vision and time-series models, Vision-Language Model (VLM)-based prompt learning and cross-modal distillation introduce significantly higher computational and runtime requirements. In particular, CLIP-like Vision-Language Models are substantially heavier than lightweight single-modality models, such as MobileNet, because they combine visual and textual encoders and require cross-modal alignment during training.

As a result, these approaches were considered less suitable for the final integrated RAIDO workflow and available platform runtime environment. Consequently, the final integrated component focuses on platform-suitable knowledge distillation workflows for vision and time-series models, while prompt learning and cross-modal distillation remain available as standalone code-level tools and experimental implementations.

The final implementation therefore centers on DKD-based vision model optimization and PatchTST-based time-series distillation workflows, providing efficient model adaptation, reduced computational requirements, and energy-efficient AI training capabilities fully integrated within the RAIDO optimization ecosystem.

Since D2.3, the Data Enrichment, Curation, Distillation and Federated Mining component has evolved from its initial design into a fully implemented solution. In its final version, the component is delivered as three complementary tools developed within T3.1, each provided for both time-series and image data: (i) the Data Preprocessing Pipeline, (ii) the AutoAnnotate tools, and (iii) the Federated Data Mining tools. All three

(3) follow a common engineering approach aligned with RAIDO's modular, Orchestrator-driven design: self-contained and independently selectable functionalities, exposed through programmatic interfaces and orchestrated as workflow tasks, implemented in Python using widely adopted scientific computing and deep learning libraries. This section summarizes the main additions and improvements introduced since D2.3 and the final implementation status of each tool.

Design rationale and deviations from the initial architecture

In line with this evolution, two of the three tools, AutoAnnotate and Federated Data Mining, were deliberately developed as standalone, self-contained tools rather than additional steps embedded within the linear preprocessing pipeline. This decision reflects their distinct execution models: whereas the Data Preprocessing Pipeline performs an automated, sequential transformation of a single dataset, AutoAnnotate is an interactive, human-in-the-loop annotation workflow, and Federated Data Mining is an inherently distributed process that operates across multiple participants and privacy boundaries. Implementing them as independent tools, each exposing its own user and programmatic interfaces, keeps every component cohesive and easier to deploy, scale, maintain, and test in isolation, allows each to be reused both within and beyond a single RAIDO optimization pipeline, and accommodates their differing runtime requirements, an interactive labelling interface for AutoAnnotate and a federated, multi-node deployment for Federated Data Mining, while still allowing the Orchestrator to invoke each capability on demand.

The final design also deviates in some respects from the architecture initially proposed in D2.3. Most notably, the Federated Data Mining module, originally envisaged primarily as a mechanism for retrieving suitable samples from other users' datasets or, where access was restricted, for forwarding metadata to the Synthetic Data Generation component, was refined into a federated dataset-distillation approach. This refinement was adopted because it more directly satisfies RAIDO's privacy-by-design and data-efficiency objectives: raw data is never exchanged or centralized, as only model parameters and gradients are shared, and the module itself produces compact synthetic data of comparable predictive value, thereby reducing data-transfer requirements and the dependence on a separate generation step. In addition, the AutoAnnotate tool was introduced to address the availability of labelled data, a practical prerequisite for supervised model training that was not explicitly covered by the two-module structure described in D2.3.

Data Preprocessing Pipeline

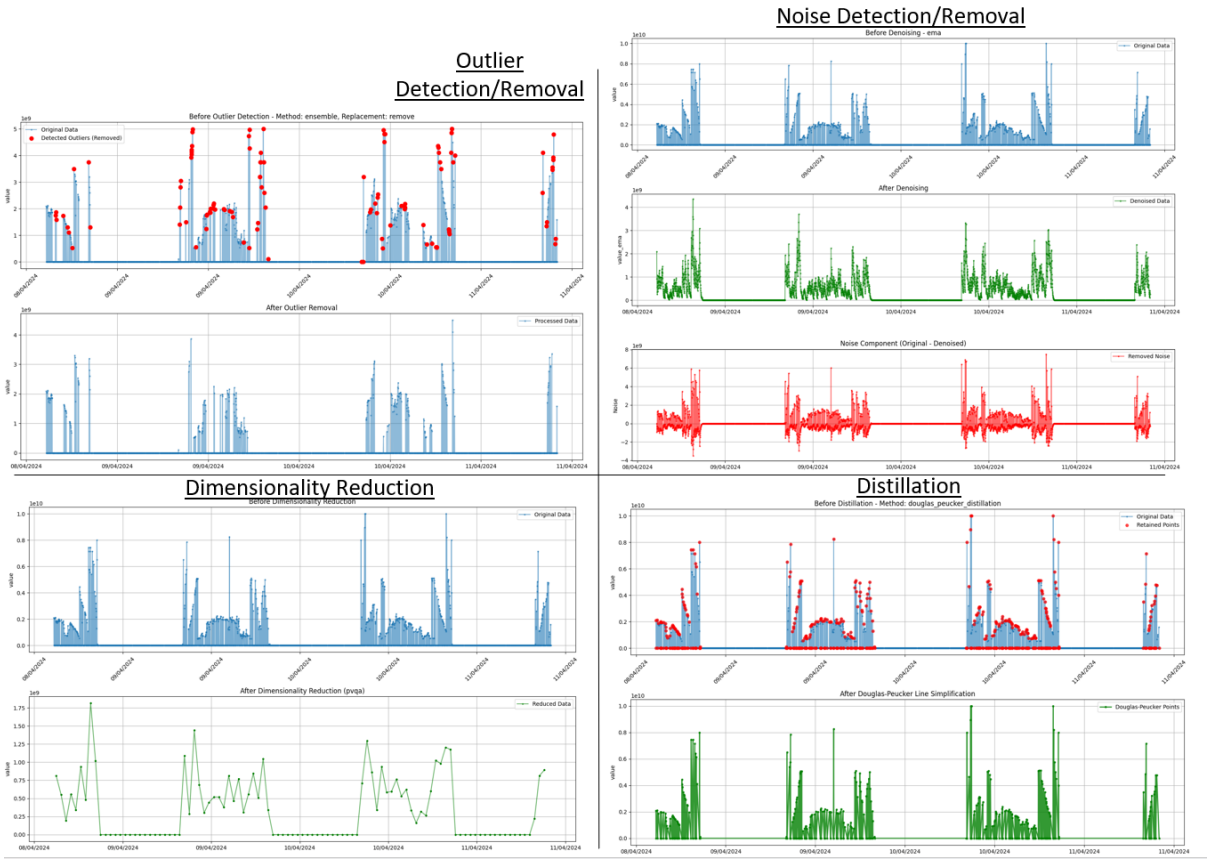

Figure 7: Example outputs of time-series preprocessing steps.

The data preprocessing functionality, introduced in D2.3 as the Data Enrichment, Curation, and Distillation module, has been fully implemented as two parallel pipelines, one for time-series and one for image data. Each pipeline is structured as a sequence of grouped functionalities that the Orchestrator can compose into a preprocessing flow tailored to the dataset and task at hand. For time-series data, these groups cover input dataset cleanup, missing-data imputation, outlier detection and treatment, noise removal, feature engineering, data balancing and enrichment, advanced feature engineering and normalization, dimensionality reduction, and dataset distillation (some examples can be seen in Figure 7). For image data, they cover invalid-pixel detection and handling, image-level outlier detection, denoising, data augmentation and transformation, deep feature extraction, class balancing, dimensionality reduction, and dataset distillation. The tool was developed in collaboration with ADRESTIA.

Each group offers multiple selectable methods. For example, several statistical and learning-based strategies for outlier detection, a range of smoothing filters for noise removal, pre-trained convolutional and transformer models for image feature extraction, and complementary techniques that distil a dataset into a compact yet representative subset. Compared with D2.3, the pipelines have matured into an operational service: individual steps are executed as orchestrated workflow tasks using Apache Airflow, the functionality can be exposed through a FastAPI service (optional), and the configuration

and execution state of each run are tracked in structured JSON files, while transformed datasets, accompanying metadata, and optional before/after visualizations are produced for downstream use within RAIDO. The input and output requirements of each step have been consolidated and documented, and the implementation is accompanied by a dedicated automated test suite.

AutoAnnotate Tools

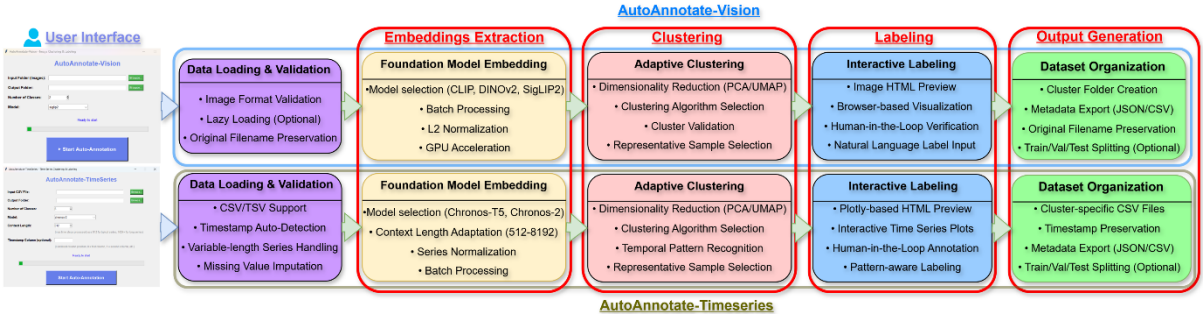

Figure 8: Common five (5) stage architecture of the AutoAnnotate tools.

A significant addition since D2.3 is the AutoAnnotate tool, which supports the semi-automatic annotation of unlabeled datasets and is provided as two sister tools: AutoAnnotate-Vision for images and AutoAnnotate-Timeseries for time-series. Both tools are a common five-stage architecture (Figure 8): (i) data loading and validation, (ii) embedding extraction using pre-trained foundation models, (iii) adaptive clustering, (iv) interactive labeling, and (v) output generation. Representative feature embeddings are extracted with state-of-the-art vision-language models and time-series foundation models, after which semantically or structurally similar samples are grouped using traditional clustering algorithms. The user is then shown representative samples from each cluster to which they assign a custom annotation, and labeled data are then organized into per-class folders, together with accompanying metadata and train/validation/test splits. Each tool can be either accessed through a CLI or GUI. These tools have additionally been disseminated through a scientific paper that was accepted and presented in IEEE-CAI 2026 in May[1].

Federated Data Mining

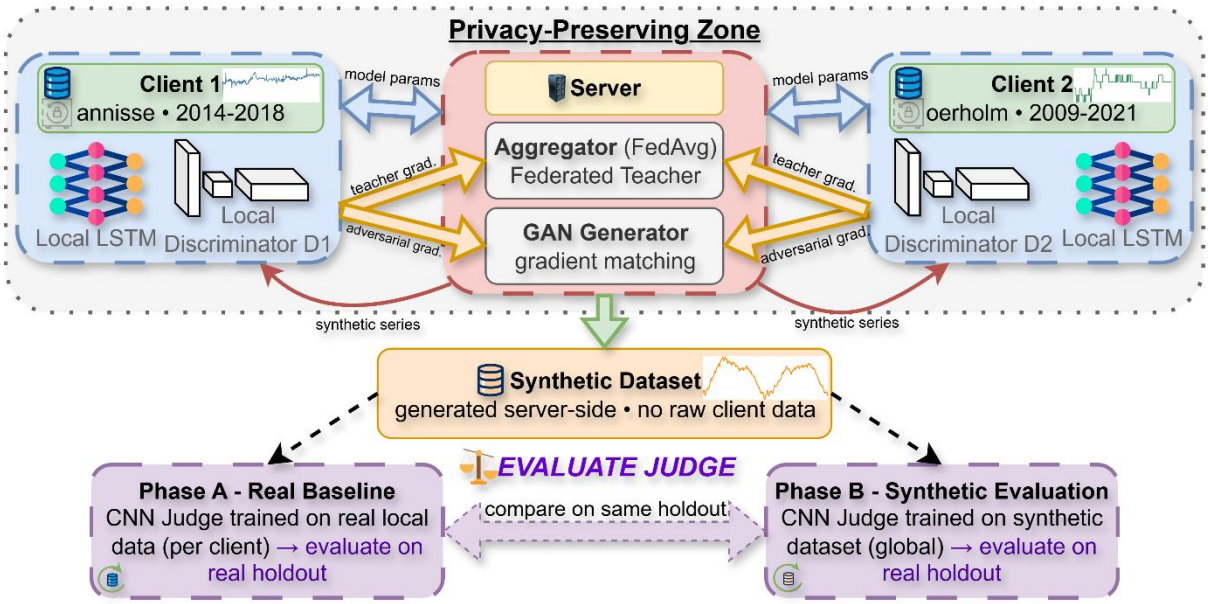

Figure 9: Privacy-preserving federated dataset distillation architecture.

The Federated Data Mining module has been finalized as a pair of privacy-preserving tools, one for time-series and one for image data, that address the dataset scarcity and imbalance by leveraging data distributed across multiple RAIDO users without ever exchanging their raw data. In its final implementation, the module adopts a federated dataset distillation approach (Figure 9): a shared teacher model is first trained collaboratively across the participating clients using federated averaging, so that raw data remains local and only model parameters and gradient information are exchanged; a generator then produces a compact set of synthetic, representative samples by matching the gradients of a student model to those aggregated from the clients, while local discriminators provide an adversarial signal that improves the realism of the generated data. For the image case, differential-privacy mechanisms can additionally be applied to the exchanged gradients to further strengthen privacy. The resulting synthetic datasets preserve the predictive value of the original distributed data at a substantially reduced volume, with their quality being validated using independent evaluation ("judge") models, which attained predictive performance close to that obtained with the full data. The development of both Federated Data Mining tools is complete, in collaboration with IITKGP, and the approach has been disseminated through a scientific paper accepted at IEEE-CH2026, which will be held in Venice in September.

In summary, the component has reached its final implementation state, with all three tools operational across the time-series and image modalities and integrated into RAIDO through programmatic interfaces and workflow orchestration, in line with the component's interactions with the Orchestrator and the Data Lake described in the preceding sections. Comprehensive technical details, including specific methods, models, parameters and evaluation results of each tool, will be reported in deliverable D3.1 (M33, T3.1).

3.5 Life-Long Learning in Federated Architectures

3.5.1 Purpose and role within RAIDO

The Life-Long Learning in Federated Architectures (FEDCL) component constitutes the second part of the AI Model Optimisation Layer and supports learning workflows within distributed and federated environments. The component addresses scenarios where data are decentralized across multiple clients or where time-series data become progressively available and need to be analyzed under changing data conditions.

FEDCL combines Federated Learning (FL) and Continual Learning (CL) capabilities. Federated Learning enables collaborative model training across multiple decentralized clients by exchanging model updates instead of raw datasets, thereby supporting privacy-preserving and distributed training. Continual Learning supports time-series learning scenarios where data evolve over time and where learning strategies must account for changing data distributions and recurring temporal patterns.

Within the RAIDO platform, FEDCL operates under the coordination of the Orchestrator and executes learning workflows using the required datasets, models, and configuration parameters. The resulting trained models, logs, reports, prediction outputs, and evaluation metrics can be stored in DATALAKE and made available to the platform for reporting, monitoring, and further analysis.

The component contributes directly to the trustworthy and Green AI objectives of RAIDO by supporting distributed model optimization, adaptive learning, privacy preservation, and efficient model evolution throughout the AI lifecycle.

3.5.2 Architecture and design principles

The initial architecture of FEDCL was designed around two complementary learning paradigms: Federated Learning and Continual Learning. For Federated Learning, the architecture was based on Flower AI[1], providing a framework for decentralized model training across distributed clients. The design envisioned support for baseline federated learning algorithms such as FedAvg[2] as well as more advanced approaches such as FedDF[3]. To support trustworthy and privacy-preserving AI, the architecture additionally

[1] *Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., ... & Lane, N. D. (2020). Flower: A friendly federated learning research framework. arXiv preprint arXiv:2007.14390.*

[2] *McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). Pmlr.*

[3] *Lin, T., Kong, L., Stich, S. U., & Jaggi, M. (2020). Ensemble distillation for robust model fusion in federated learning. Advances in neural information processing systems, 33, 2351-2363.*

considered differential privacy techniques[4], secure aggregation protocols[5], and fairness-aware learning approaches.

For Continual Learning, the architecture initially relied on the Avalanche framework and was designed to support lifelong learning workflows capable of continuously integrating new information while preserving previously acquired knowledge. The objective was to mitigate catastrophic forgetting and enable continuous model adaptation under evolving data conditions.

Together, these capabilities established the foundation for a flexible learning framework capable of supporting distributed training, privacy preservation, continual adaptation, and long-term model evolution across the RAIDO ecosystem.

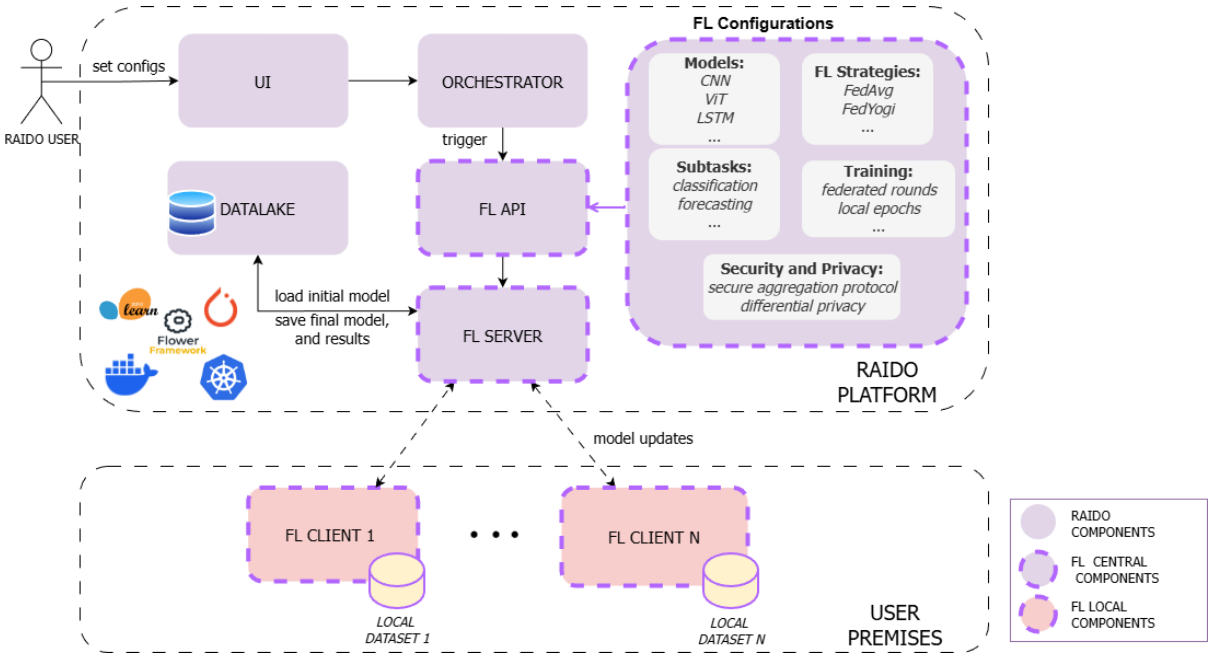

Figure 10: Federated & Continual Learning component architecture.

3.5.3 Interactions with other components

FEDCL operates as part of the RAIDO platform back-end and is triggered through ORCH. The Orchestrator forwards user-defined configurations, datasets, model parameters, and execution settings to the selected Federated Learning or Continual Learning workflow, initiates execution, and receives the final outputs generated by the component.

The component communicates directly with DATALAKE throughout its execution lifecycle. For Federated Learning workflows, this includes retrieving initial model

[4] *Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 308-318).*

[5] *Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., ... & Seth, K. (2016). Practical secure aggregation for federated learning on user-held data. arXiv preprint arXiv:1611.04482.*

parameters, updating training progress, and storing the final global model together with the corresponding performance, energy, and fairness reports. For the Continual Learning workflow, DATALAKE provides access to the required time-series datasets and execution parameters. Upon completion, the workflow produces a trained model and a JSON results file containing prediction outputs and evaluation metrics, which can be stored within DATALAKE and used by the platform for reporting and further analysis.

FEDCL additionally consumes datasets generated through ENRICH and DATAGEN and produces optimized models that can subsequently be analyzed through XAI, benchmarked through RLFEED, and deployed through ORCH and RSCMANAGER.

3.5.4 Additions, improvements, and final version of the component

Since D2.3, the Federated Learning part of the component has evolved into a fully integrated platform service. The implementation builds on Flower AI combined with Kubernetes, Docker, PyTorch, and Scikit-Learn and provides a configurable and extensible framework for training models on decentralized private datasets. It supports both baseline federated learning algorithms such as FedAvg and advanced approaches such as FedDF, together with a broad suite of machine learning models covering multiple data modalities and application domains.

The final implementation incorporates global model distillation, learning-rate and epoch-decay strategies, model parameter regularization, client-drift mitigation mechanisms, differential privacy techniques, secure aggregation protocols, and fairness-aware learning approaches. The workflow additionally supports automatic client orchestration, execution monitoring, logging, and result collection, while exposing APIs for integration with ORCH and seamless execution through the RAIDO user interface.

In the final version of the Federated Learning component, novel Federated Continual Learning strategies were additionally developed by combining Federated Learning and Continual Learning paradigms. Specifically, the FedCurv-DR strategy was implemented in Flower AI leveraging an adaptation of Elastic Weight Consolidation (EWC)[6]. It enables continuous model adaptation on streams of image datasets through model parameter regularization while maintaining communication efficiency, computational efficiency, and fairness-aware learning capabilities.

Similarly, FLwF*[7] was implemented by combining the FedAvg baseline with Learning without Forgetting (LwF)[8], leveraging knowledge distillation techniques to mitigate

[6] *Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences, 114(13), 3521-3526.*

[7] *Usmanova, A., Portet, F., Lalanda, P., & Vega, G. (2022, June). Federated continual learning through distillation in pervasive computing. In 2022 IEEE International Conference on Smart Computing (SMARTCOMP) (pp. 86-91). IEEE.*

[8] *Li, Z., & Hoiem, D. (2017). Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 40(12), 2935-2947.*

catastrophic forgetting during local model training. Both approaches contribute directly to the sustainability objectives of RAIDO and align with Green AI principles.

In addition to the Federated Continual Learning strategies described above, the final version of the component includes a dedicated time-series Continual Learning workflow. Compared with the initial architecture described in D2.3, where Avalanche was considered for Continual Learning, the final implementation was refined to better match the data type, workflow requirements, and integration constraints of the RAIDO platform.

During the finalization of the component, the available Avalanche-based implementation was found not to directly support the time-series Continual Learning workflow required for the integrated RAIDO use case. Adapting it to the final time-series setting would require additional engineering effort, including modifications to data handling procedures, model interfaces, evaluation outputs, and platform integration mechanisms. Therefore, it was not selected as the basis for the final integrated Continual Learning workflow.

Instead, the final version adopts an FSNet-based implementation specifically aligned with time-series Continual Learning. This implementation is better suited to sequential time-series data, changing data distributions, and recurring temporal patterns while also providing a more practical basis for integration into the RAIDO workflow.

In the current integrated version, the workflow consumes a time-series dataset provided in the required tabular format and produces a trained model together with a JSON results file. The JSON file records prediction outputs and the corresponding evaluation metrics, including Mean Absolute Error (MAE) and Mean Squared Error (MSE). These outputs can be stored within DATALAKE and made available to the RAIDO platform through ORCH for reporting, monitoring, and further analysis.

The final FEDCL component therefore combines a fully integrated Federated Learning service, Federated Continual Learning extensions, and an FSNet-based Continual Learning workflow, supporting distributed training, adaptive learning, privacy preservation, and long-term model evolution across the RAIDO pilot environments.

3.6 Data Lake

3.6.1 Purpose and role within RAIDO

The Data Lake constitutes the core information management layer of the RAIDO platform and serves as the central repository for storing, managing, retrieving, and governing all datasets, AI models, metadata, monitoring information, and workflow artefacts generated throughout the AI lifecycle. As one of the foundational components of the RAIDO architecture, the Data Lake enables interoperability between all platform services and provides the storage infrastructure upon which the remaining optimization, orchestration, explainability, monitoring, and deployment components operate.

The business logic of the component is centered on providing a unified, scalable, and secure environment capable of supporting the highly heterogeneous data requirements of the RAIDO pilots. These requirements range from structured tabular datasets and time-series measurements to high-resolution images, simulation outputs, healthcare records, AI models, execution artefacts, monitoring telemetry, and energy consumption metrics. By centralizing these resources within a common information layer, the Data Lake simplifies data management, enables data reuse, supports reproducibility, and facilitates efficient AI model development and deployment.

In addition to its storage capabilities, the Data Lake contributes directly to the realization of RAIDO's Green AI objectives by supporting hardware utilization monitoring, energy consumption tracking, model lifecycle management, and reusable AI workflows. Through integrated security, governance, and access control mechanisms, the component ensures that data and AI artefacts remain protected, traceable, and accessible only to authorized users and services.

3.6.2 Architecture and design principles

The initial Data Lake architecture presented in D2.3 was designed as the information layer of the RAIDO platform, responsible for storing datasets and AI models while supporting efficient AI training and deployment processes. The architecture adopted a centralized storage philosophy capable of handling structured, semi-structured, unstructured, and time-series data originating from multiple pilots and application domains.

The original design proposed the integration of four specialized database technologies, each optimized for a specific category of information. MySQL was selected for structured relational and tabular data, MongoDB for semi-structured and document-oriented datasets, MinIO for AI models and large binary artefacts, and InfluxDB for time-series information. This approach allowed each data modality to be managed by the most appropriate storage technology while maintaining a unified access layer across the platform.

The architecture was additionally designed to support data governance, security, metadata management, and interoperability between platform components. Scalability, flexibility, and efficient storage management were identified as key design principles, enabling the infrastructure to accommodate the diverse requirements of the RAIDO pilots while supporting future expansion.

The high-level architecture of this component is shown in Figure 11.

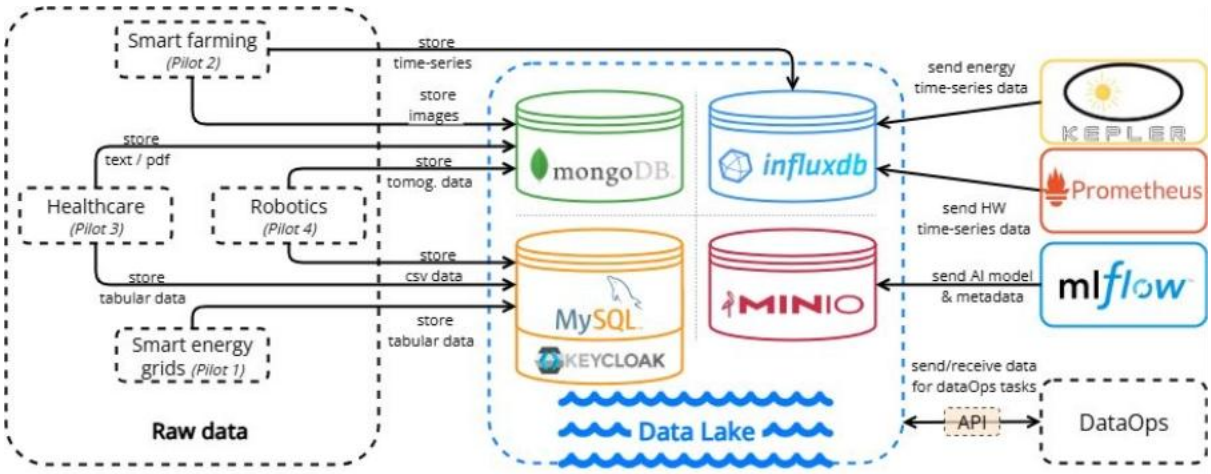

Figure 11. Data Lake component architecture.

3.6.3 Interactions with other components

The Data Lake interacts with virtually all components of the RAIDO platform and serves as the central information exchange mechanism between them.

ORCH utilizes the Data Lake to retrieve datasets, store workflow metadata, access trained models, and manage the execution of optimization pipelines. ENRICH and DATAGEN retrieve raw datasets from the Data Lake and subsequently store transformed, enriched, balanced, or synthetic data for reuse by downstream workflows. DISTIL and FEDCL access training datasets, store optimized models, retrieve historical artefacts, and maintain model versioning information throughout the optimization lifecycle.

XAI and RLFEED utilize the Data Lake to access models, datasets, evaluation results, monitoring information, and feedback artefacts required for explainability analysis, benchmarking activities, and optimization monitoring. The Visualization Engine interacts extensively with the Data Lake through the backend services, allowing users to upload datasets, retrieve results, visualize monitoring information, access trained models, and manage AI workflows through a unified interface.

The component also interacts with the platform monitoring infrastructure. Prometheus, Kepler, InfluxDB, and Grafana exchange monitoring and telemetry information through the Data Lake ecosystem, supporting hardware utilization monitoring, energy consumption analysis, observability, and Green AI evaluation activities.

3.6.4 Additions, improvements, and final version of the component

Since Deliverable D2.3, the Data Lake has evolved from a conceptual storage architecture into a fully implemented, deployed, and operational information management platform. While the fundamental architectural principles remained unchanged, the final implementation significantly expanded the original design and transformed it into a complete production-ready infrastructure supporting all RAIDO platform services.

A major enhancement compared to the initial architecture is the implementation of a polyglot persistence strategy. Rather than relying solely on the four storage technologies originally proposed, the final platform integrates a broader ecosystem of complementary technologies, including MySQL, MongoDB, MinIO, InfluxDB v1, InfluxDB v2, Prometheus, Kepler, Grafana, FastAPI, Keycloak, and Kubernetes. Each technology addresses specific storage, monitoring, analytics, security, or orchestration requirements while remaining fully integrated within a unified backend architecture.

The final implementation supports all data modalities required by the project, including tabular datasets, time-series measurements, images, documents, AI models, execution artefacts, monitoring telemetry, and energy-consumption metrics. A dedicated technology mapping has been established, where MySQL manages structured relational data, MongoDB stores images and document-oriented artefacts, MinIO provides object storage for AI models and large binary files, InfluxDB v1 stores hardware monitoring telemetry, and InfluxDB v2 stores energy-consumption information generated through Kepler.

The component has also been fully integrated into the Kubernetes-based deployment infrastructure of RAIDO. Each storage service operates as an independent microservice, enabling fault isolation, scalability, maintainability, and flexible deployment across both cloud and on-premises environments. This architecture supports the diverse operational requirements of all pilots, including GDPR-compliant on-premises deployments for healthcare-related use cases.

Significant improvements were additionally introduced in terms of security and governance. Authentication and authorization are implemented through a centralized Keycloak-based identity management infrastructure, providing role-based access control, token management, and secure interaction between users, services, and platform components. The backend infrastructure exposes a comprehensive set of REST APIs implemented through FastAPI, enabling seamless integration with the Visualization Engine, AI workflows, monitoring services, and DataOps processes.

The final version of the Data Lake therefore represents considerably more than a storage repository. It operates as the central information backbone of the entire RAIDO platform, supporting data management, AI model lifecycle management, workflow integration, monitoring, observability, energy-awareness, security, governance, and interoperability across all project pilots and platform services.

3.7 Frameworks for AI explainability

3.7.1 Purpose and role within RAIDO

The Explainable Artificial Intelligence (XAI) component constitutes the core of the Trustworthy AI Layer within the RAIDO platform and is responsible for ensuring

transparency, fairness, robustness, accountability, and trustworthiness throughout the AI lifecycle. The component aligns with the overall RAIDO objective of developing reliable and human-centric AI solutions that comply with the principles of trustworthy AI and the European approach to Artificial Intelligence.

The XAI framework is designed to evaluate both the quality of the input data and the behavior of the resulting AI models. Through a combination of explainability techniques, bias detection mechanisms, fairness assessment methodologies, and trustworthiness indicators, the component enables users to understand how AI systems reach their decisions and whether these decisions satisfy predefined ethical and operational requirements.

The component serves as a horizontal service accessible by all AI and data-processing modules of the RAIDO ecosystem. It provides mechanisms for evaluating data quality, detecting dataset and model biases, measuring fairness, generating explanations, supporting human oversight, and ensuring compliance with trustworthy AI principles. In doing so, the component contributes directly to increased transparency, improved user confidence, enhanced Human-in-the-Loop (HITL) acceptance, and the delivery of more reliable AI solutions across all project pilots.

3.7.2 Architecture and design principles

The initial architecture of the XAI component was introduced in Deliverable D2.3 as a dedicated Trustworthy AI Layer operating through two primary checkpoints positioned at different stages of the AI lifecycle.

The first checkpoint was designed to evaluate the quality and trustworthiness of input data before entering the optimization pipeline. This checkpoint focused on examining data sources, assessing transparency and reliability, evaluating data collection and annotation procedures, detecting biases, identifying privacy concerns, and validating data preprocessing activities. Particular emphasis was placed on ensuring that data enrichment, balancing, curation, and novelty discovery procedures complied with ethical and fairness requirements.

The second checkpoint was positioned after the optimization process and focused on evaluating trained AI models. This stage assessed learning algorithms, performance metrics, model variance, robustness, and fairness characteristics while identifying potential sources of bias. The objective was to ensure that the optimized AI models generated by the platform satisfied predefined trustworthiness criteria before being deployed or presented to users.

To support these objectives, the original architecture incorporated several state-of-the-art explainability methodologies. SHapley Additive Explanations (SHAP) were adopted to provide global feature attribution and quantify the contribution of individual features to model predictions. Local Interpretable Model-Agnostic Explanations (LIME) were

introduced to generate localized explanations capable of explaining individual predictions. Additional explainability techniques included Layer-wise Relevance Propagation (LRP), counterfactual explanations, and Gradient-weighted Class Activation Mapping (Grad-CAM), providing complementary perspectives on model behavior and decision-making processes.

The architecture additionally incorporated bias detection and ethics evaluation mechanisms based on latent-space analysis of trained models. These mechanisms were designed to identify discriminatory behavior, unfair treatment of demographic groups, and other ethical concerns. Lightweight blockchain-based mechanisms were also envisaged to support accountability, traceability, and certification of AI processes throughout the platform.

Collectively, these architectural elements established the foundations of a comprehensive explainability and trustworthiness framework capable of supporting all AI workflows within the RAIDO ecosystem.

The high-level architecture of this component is shown in Figure 12.

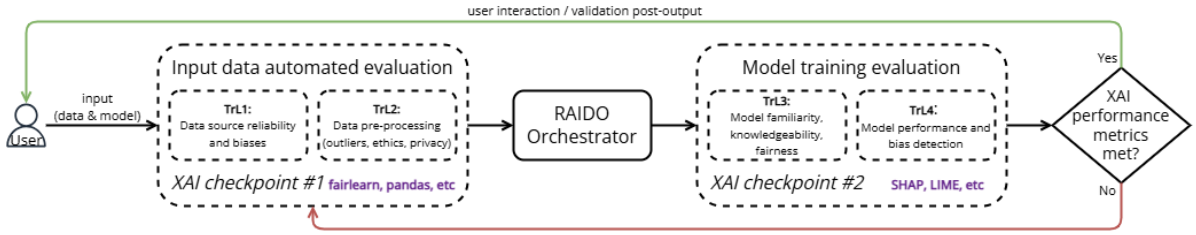

Figure 12. XAI component: Internal architecture.

3.7.3 Interactions with other components

The XAI component operates as a cross-cutting service that interacts with virtually all components of the RAIDO platform and acts as the primary trustworthiness assessment mechanism throughout the AI lifecycle.

Its first interaction occurs with ENRICH, where XAI evaluates the quality, representativeness, and fairness of input datasets before they enter the optimization pipeline. Data quality inspections, bias assessments, privacy evaluations, and trustworthiness analyses are performed at this stage to ensure that only validated datasets are utilized by downstream components.

XAI receives datasets, AI models, and prediction results from the optimization workflow to generate explainability artefacts, fairness assessments, and trustworthiness metrics. These outputs are stored in the DATALAKE and made available to ORCH, RLFEED, and the Visualization Engine for subsequent evaluation and presentation.

The component additionally interacts with DATAGEN, DISTIL, and FEDCL, evaluating the outputs generated by these services and assessing the impact of synthetic data generation, model distillation, federated learning, and continual learning on fairness, robustness, and explainability.

All datasets, trained models, explainability artefacts, fairness reports, trustworthiness metrics, and generated visualizations are retrieved from and stored within the DATALAKE, enabling traceability, reproducibility, and lifecycle management of explainability analyses.

The component further interacts with the platform's Visualization Engine and user interface services, providing interpretable outputs, visual explanations, fairness assessments, trustworthiness indicators, and detailed analytical reports that support informed decision-making by end users and domain experts.

Through these interactions, XAI establishes a continuous trustworthiness assessment layer capable of evaluating both data and AI models throughout the complete optimization lifecycle.

3.7.4 Additions, improvements, & final version of component

The overall architecture and the core logic proposed in the initial stages of the project remain valid and largely intact. The overarching goal of the RAIDO XAI Engine—to establish a Trustworthy AI Layer via two main checkpoints evaluating both input data and optimized models—drove the component's development. The additions and improvements summarized below represent how the original prototype component evolved, while more details are presented in D4.2.

A primary advancement in the final component is the heavy integration of counterfactual explanations. User studies consistently show that user satisfaction and trust are significantly improved by actionable explanations and justifications. One major issue in XAI is the degree to which users can understand the cause of a decision without needing a background in data science. Counterfactual explanations solve this by showing how the AI model's decision outcome would have been different under different input feature conditions (e.g., "If you had asked for a lower loan, your application would have been approved"). This approach bypasses the technical challenges of explaining complex algorithmic rules, providing a minimal amount of information necessary for a decision to be changed, and directly satisfying legal requirements for algorithmic recourse in automated decisions.

The Four Trust Levels (TrLs) Architecture

As presented before, to operationalize the two original checkpoints, the finalized XAI engine operates on the following four distinct Trust Levels (TrLs). The first two TrLs constitute Checkpoint #1 at the RAIDO input stage, assessing the quality and

representativeness of the input data. The latter two TrLs constitute Checkpoint #2, assessing the fairness and explainability of the model predictions.

- TrL1 (Data Quality Inspection): Examines data source trustworthiness by checking for quality issues, missing values, duplicates, outliers, and privacy risks.
- TrL2 (Data Bias Detection): Detects representation bias in input data by assessing sensitive attribute distributions, class imbalances, and temporal drift.
- TrL3 (Model Fairness Assessment): Evaluates fairness across different demographic groups by measuring performance disparities, error correlations, and equalized odds.
- TrL4 (Model Explainability): Unveils the model's decision-making logic using global feature importance, local explanations, counterfactuals, and visual saliency maps.

XAI on Tabular Data

For structured, tabular data, the XAI engine workflow operates in two distinct modes. The "Training Mode" requires raw historical data to fit models and establish baseline statistics, while the "Inference Mode" ingests batches of new, unseen data for real-time assessment. The framework is model-agnostic, supporting datasets with or without a temporal dimension, and is capable of handling both regression and classification tasks.

During TrL1 (Data Inspection), each tabular dataset undergoes a rigorous eight-stage automated validation pipeline. This includes a header check to remove index artifacts, duplicate row isolation, an exact missing value assessment, and data formatting validation. It also executes a zero-variation check to flag constant columns, an outlier detection scan applying the 1.5 * IQR rule, a chronological time-series integrity check, and a privacy risk assessment that flags columns with >90% uniqueness as potential Personally Identifiable Information (PII).

For TrL2 (Data Bias Detection), the system evaluates representation bias to prevent discriminatory model behavior. The primary framework analyzes the distribution of user-defined sensitive features. For categorical attributes, it highlights minority classes; for numerical attributes, it evaluates skewness and triggers warnings for density ranges containing less than 5% of the data. If a time column is present, it performs a temporal representation analysis to track distribution drift and yearly imbalances. For specific, human-centric tabular datasets, the user can select an alternative option. This layer utilizes a Composite Balance Score (CBS), a specialized metric adapted for multi-class protected attributes and intersectional fairness, allowing for a more comprehensive bias assessment.

Moving to the model output stage, TrL3 evaluates algorithmic fairness by measuring the consistency of model performance on an unseen test dataset. To prevent artificially optimistic results caused by overfitting, TrL3 dynamically bins continuous sensitive

attributes and calculates disaggregated metrics using the Fairlearn library. For regression, it computes the Mean Squared Error (MSE) Disparity Ratio between the worst- and best-performing demographic groups. For classification, it calculates disparities in Accuracy, Precision, Recall (Equal Opportunity), and Selection Rate (Demographic Parity), applying strict statistical risk thresholds to flag unfairness.

Finally, TrL4 provides insights on the complex model decisions. It executes a three-stage explainability pipeline: First, it utilizes SHAP (Shapley Additive Explanations) to provide global explainability, isolating the top driving features of the model. Second, it employs LIME (Local Interpretable Model-agnostic Explanations) for localized inference, generating 5,000 perturbed data points to explain specific single-row predictions. Third, it uses DiCE (Diverse Counterfactual Explanations) to generate actionable "what-if" scenarios, providing users with three different pathways to flip a classification prediction or get distinct standard deviation shifts for regression outcomes.

XAI on Image Data

To handle computer vision tasks, the four TrL modules are adapted to process image directories and PyTorch models (.pth files), natively supporting architectures like ResNet18, ResNet50, and AlexNet.

In TrL1, the system performs a physical integrity check on the raw image files (.jpg, .jpeg, .png). This six-stage pipeline rejects corrupted files smaller than 100 bytes, verifies structural headers, forces a full pixel-data load to catch truncated files, ensures minimum pixel dimensions, checks format consistency to catch spoofed extensions, and tests metadata accessibility.

TrL2 assesses the image dataset for representation bias by calculating the Imbalance Ratio (IR) across class subdirectories. Predefined thresholds evaluate the risk: an IR below 1.5 is considered low risk, while an IR above 3.0 indicates severe imbalance, warning data scientists that the unweighted deep learning model will likely degrade and heavily bias its predictions toward the majority class. The system also visually plots the deviation of each class's percentage share against an expected uniform distribution baseline.

In TrL3, algorithmic fairness is evaluated by running inference on a previously unseen test dataset to build a vectorized confusion matrix. The module computes disaggregated, per-class metrics (Accuracy, Precision, Recall, F1-Score) alongside aggregate fairness indicators such as Equalized Odds, Equal Opportunity, and Predictive Equality. The statistical significance of any performance differences across image classes is rigorously assessed using a one-way ANOVA test.

For explainability (TrL4), the engine shifts from feature importance to visual mapping. It generates saliency maps using RISE (Randomized Input Sampling for Explanation). This

gradient-free perturbation method generates random binary masks, upsamples them, and runs masked images through the model. The resulting prediction scores are aggregated into a normalized heatmap overlay (using a blue-to-red scale), clearly highlighting the specific pixel regions that most heavily influenced the AI's classification.

Integration into the RAIDO Platform

To ensure seamless, highly responsive operation within the wider architecture, the RAIDO XAI Engine is deployed as a self-contained, stateless FastAPI microservice containerized via Docker. Sitting alongside the RAIDO Data Lake, it utilizes internal cluster networking to securely fetch datasets and models.

To optimize performance and avoid burdening the client UI with large network payloads, the API operates in a "Fetch Mode." It does not accept direct file uploads (such as massive CSVs or PyTorch models). Instead, the client UI makes a lightweight POST request to one of the eight core execution endpoints (e.g., /trl4/tabular/execute/{user_id}), passing only the user ID and relevant dataset/model IDs. The container then securely queries the Data Lake API, downloads the files into an isolated temporary directory, and executes the heavy analytical audit in-memory.

Once the analysis is complete, the platform generates a comprehensive, highly visual PDF report, which is posted back to the Data Lake. The container then automatically signals the RAIDO Orchestrator to update the pipeline step status to "Succeeded", returns a lightweight JSON confirmation to the frontend, and instantly deletes all ephemeral data to free up memory.

3.8 Reinforced benchmarking and feedback-based progress monitoring

3.8.1 Purpose and role within RAIDO

The Reinforced Benchmarking and Feedback-Based Progress Monitoring component, referred to as RLFEED within the final RAIDO platform, constitutes the feedback and optimization layer of the RAIDO ecosystem. The component is responsible for continuously evaluating the outcomes of optimization workflows, benchmarking AI models against predefined objectives, and recommending improved configurations capable of enhancing performance, efficiency, and sustainability.

The business logic of the component is based on the concept of closed-loop optimization. Rather than treating AI optimization as a single execution process, RLFEED introduces an iterative feedback mechanism through which optimization results are continuously monitored, evaluated, and improved. The component receives operational metrics, model parameters, and performance indicators from the optimization

workflows and uses these inputs to propose improved configurations that can be re-executed by the orchestration layer.

The original vision of the component was based on multi-objective Reinforcement Learning and active learning methodologies capable of simultaneously optimizing several competing objectives such as model performance, execution duration, explainability, and energy efficiency. To increase transparency and trustworthiness, the component was additionally designed to incorporate Human-in-the-Loop (HITL) supervision and feedback-driven learning mechanisms capable of influencing optimization decisions.

Through these capabilities, RLFEED contributes directly to the realization of the RAIDO Green AI objectives by enabling continuous optimization of AI workflows while balancing quality, computational cost, and energy consumption across heterogeneous deployment environments.

3.8.2 Architecture and design principles

The initial architecture of the component was introduced in D2.3 as a reinforcement-based optimization framework positioned after the AI optimization layer. The architecture was designed to operate as a dynamic evaluation and optimization mechanism capable of continuously improving AI workflows through iterative benchmarking and feedback analysis.

The original design adopted multi-objective optimization principles, where several Key Performance Indicators (KPIs) were simultaneously considered during optimization. These KPIs included performance-oriented metrics, efficiency indicators, explainability requirements, and energy-consumption measurements. The architecture envisioned automatic KPI selection mechanisms based on Machine Learning, Deep Learning, and evolutionary optimization techniques capable of adapting evaluation criteria to different application domains and datasets.

The component was designed to receive encoded operational parameters, optimization outputs, and KPI measurements from the orchestration framework. Reinforcement Learning agents would subsequently evaluate these results, identify opportunities for improvement, and generate updated optimization parameters for future executions. Human-in-the-Loop supervision was additionally incorporated to provide oversight of automated decisions and ensure alignment with operational requirements and user expectations.

To improve scalability, the architecture also explored hierarchical optimization strategies in which lower-level optimizations performed at feature or parameter level could contribute to higher-level system optimization objectives. This design aimed to distribute optimization workloads and facilitate the emergence of complex optimization behaviors across the platform.

The resulting architecture established a flexible optimization framework capable of supporting benchmarking, feedback-driven learning, reinforcement learning, multi-objective optimization, and continuous performance improvement across the AI lifecycle.

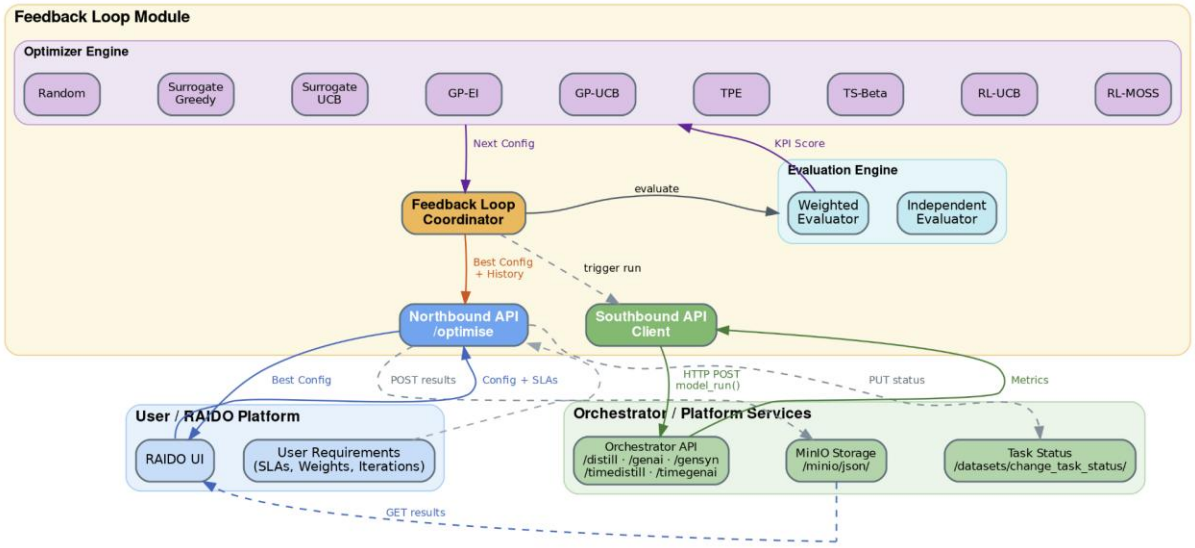

Figure 13: Feedback loop module architecture.

3.8.3 Interactions with other components

RLFEED interacts primarily with the orchestration and optimization services of the RAIDO platform and serves as the central feedback mechanism for optimization workflows.

Its closest interaction occurs with ORCH, from which it receives execution metrics, benchmarking data, KPIs and monitoring information. Following evaluation, RLFEED generates improved configurations and optimization recommendations that are returned to ORCH for subsequent execution cycles. This interaction establishes a continuous optimization loop that enables progressive refinement of AI workflows.

The component additionally interacts with DISTIL, FEDCL, DATAGEN, and other optimization services through the orchestration framework. The outputs generated by these services are benchmarked using the KPI evaluation mechanisms provided by RLFEED, allowing the platform to identify configurations that deliver improved performance, reduced execution time, or lower energy consumption.

RLFEED utilizes the DATALAKE for storing optimization artefacts, benchmarking results, KPI histories, optimization traces, and execution metadata. This enables traceability, reproducibility, and longitudinal analysis of optimization activities throughout the platform lifecycle.

The component further supports human oversight through interactions with the platform user interface and Human-in-the-Loop mechanisms. Users can review optimization

results, modify KPI priorities, adjust optimization objectives, and initiate new optimization cycles based on the generated recommendations.

Through these interactions, RLFEED establishes a continuous feedback-driven optimization process capable of improving AI workflows while supporting transparency, accountability, and sustainability objectives.

3.8.4 Additions, improvements, and final version of the component

The overall architecture and core logic proposed during the initial stages of the project remain valid and largely intact. The overarching goal of reinforced benchmarking and feedback-based progress monitoring within RAIDO was to establish a robust optimization layer that continuously evaluates KPI-driven outcomes and returns improved configurations to the orchestration pipeline. The additions and improvements summarized below describe how the initial prototype evolved into the final implementation; further implementation and validation details are reported in D4.2.

In line with the initial plan, the team implemented reinforced benchmarking strategies, including RL-based approaches, while also developing the broader optimizer portfolio (random-search baseline, surrogate-based methods, and Bayesian methods) in the same proof-of-concept environment. After systematic testing and comparative evaluation, the team concluded that Bayesian optimization performs better and more consistently under the deterministic orchestrator setting; therefore, it was selected as the default operational strategy, while all reinforced benchmarking methods remain available for advanced usage and comparative analysis.

Final Optimizer Strategy and Rationale

The initial proposal envisioned Reinforcement Learning as the primary mechanism. In the final implementation, RL was not selected as the default approach because the RAIDO orchestrator behaves deterministically: the same configuration produces the same outcome, so repeated arm-pulls do not provide new information for bandit-style learning.

For this reason, Bayesian optimization is used as the default operational strategy, as it is better suited to structured exploration in deterministic hyperparameter spaces. All nine optimization strategies, including RL-based methods, remain available for advanced usage and comparative analysis.

Final Component Interactions and Scope

The Feedback Loop Module integrates with two primary RAIDO components:

i. AI Optimization (Orchestrator): Receives operational data (model parameters, hyperparameter configurations) and KPI metrics (performance, duration, energy).

Proposes modified actions (optimized configurations) back to the Orchestrator for re-execution, creating an iterative closed-loop optimization cycle.

ii. HITL (Human-in-the-Loop): Although the full HITL mechanism is not implemented exactly as initially described, a practical HITL pattern is present in the final workflow: users review optimization outcomes and can re-run the process with updated KPI weights and parameter settings. In this way, human oversight is introduced through iterative user-guided reconfiguration, even if a dedicated real-time HITL control loop is not yet fully integrated.

Final SLO and KPI Configuration

The final implementation uses a three-KPI multi-objective setup: Performance, Duration, and Energy. In practice, users can configure the optimization policy per run, mainly through KPI weights and the maximum number of iterations.

- Configurable KPI weights: Users define the relative importance of performance, duration, and energy in weighted mode (the system normalizes weights internally).
- Configurable max iterations: Users set `iterations`, which determines the optimization budget.
- Improvement target: A weighted-goal threshold is set through goal in weighted evaluation mode.
- Baseline establishment: The first execution with the initial configuration is used as the baseline for all improvement calculations.

For the Performance KPI, the implementation supports the following quality metrics: R2, mse, mae, accuracy, precision, recall, inception_score, fid, discriminative_score, and predictive_score.

This configuration allows users to explicitly prioritize model quality, runtime, and energy efficiency according to use-case requirements, in line with RAIDO's green-AI objectives.

Final Implementation Architecture

The final Feedback Loop Module is deployed as a containerized Docker microservice (ghcr.io/axonlogic/raido-feedback-loop) using FastAPI for REST API exposure, Python with scikit-learn/NumPy/SciPy for optimization logic, and MinIO for results persistence. The Northbound interface (/optimise, `/results`) interfaces with the RAIDO UI; the Southbound client communicates with Orchestrator endpoints (distill, genai, gensyn, timedistill, timegenai). Task status updates via change_task_status complete the end-to-end integration with the platform's orchestration framework.

The operational flow during each optimization cycle proceeds as follows: the system begins with a baseline run using default hyperparameters, computes initial KPI metrics (performance, duration, energy), selects an optimizer strategy (with GP-EI as default),

evaluates all candidate configurations, and iteratively refines parameters until convergence or iteration limits are reached. The loop completes by persisting best-found configurations and metrics to MinIO, updating task status, and returning optimized parameters to the Orchestrator for validation.

3.9 Adaptive AI processes and visualization engine

3.9.1 Purpose and role within RAIDO

The Adaptive AI Processes and Visualization Engine constitutes the primary user-facing component of the RAIDO platform and serves as the central interaction layer through which users access datasets, AI models, optimization services, explainability mechanisms, monitoring information, and reporting functionalities. The component was originally conceived as a dynamic visualization framework capable of providing adaptive, user-centric visualizations suited to different user needs and application domains. Its objective is to facilitate the interaction between users and the underlying AI services while presenting complex data, model outputs, optimization results, and trustworthiness assessments through intuitive and understandable visual interfaces.

The Visualization Engine incorporates Human-in-the-Loop (HITL) support, enabling users to participate actively in AI-driven workflows. Within the final platform, this is realized primarily through guided, wizard-based task configuration and through the Reports and Analytics interface, where users review task outcomes, model performance indicators, and explainability results. The broader vision of feedback-driven optimization is supported at the design level and is exposed progressively as the underlying services mature.

Within the final RAIDO platform, the Visualization Engine evolved beyond a simple visualization framework and became the main entry point to the entire ecosystem. Through a unified web-based interface, users can manage datasets and models, configure AI workflows, execute AI services, monitor execution progress, inspect explainability results, and review generated reports. As a result, the component plays a fundamental role in ensuring usability, accessibility, transparency, and effective interaction across all layers of the RAIDO architecture.

3.9.2 Architecture and design principles

The Visualization Engine was initially designed as an adaptive visualization framework supporting data processing, AI model optimization, explainability analysis, and reporting activities across the RAIDO platform. The design was based on the principle that different stakeholders require different views of the same information and therefore visualizations should adapt dynamically according to user roles, contexts, and operational objectives.

The original design incorporated modern web technologies and envisioned support for multi-dimensional, interactive visualizations of datasets, model metrics, optimization

results, explainability outputs, and system performance indicators. A key architectural principle was the integration of Human-in-the-Loop mechanisms directly within the visualization layer, so that the component would act not only as a passive reporting tool but also as an active participant in optimization workflows by collecting user feedback, presenting recommendations, and facilitating user-driven decision making. Autonomous reporting capabilities were also included from the initial design stage, enabling the automatic generation of summaries and analytical reports suited to different stakeholder groups.

In the final implementation, the technology stack was consolidated around Vue.js 3 with the CoreUI component framework, Pinia for state management, Vue Router, and Vite as the build tool. Several capabilities envisioned in the initial design, such as Digital Twins, AI-assisted visual analytics, and context-aware adaptive dashboards, were reprioritised in favour of delivering a stable, production-ready interface covering the full portfolio of RAIDO AI services, while preserving the core principles of usability, accessibility, and transparency throughout the optimization lifecycle.

The high-level architecture of this component is shown in Figure 14.

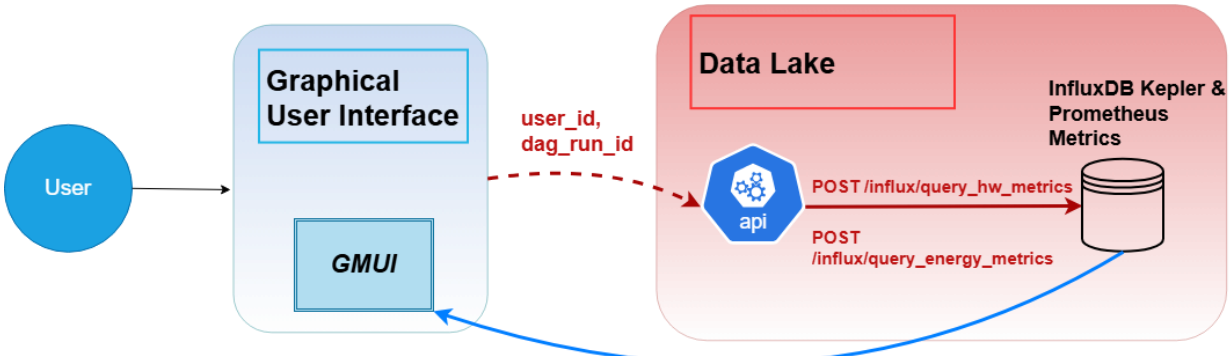

Figure 14: Green Monitoring User Interface (GMUI).

3.9.3 Interactions with other components

The Visualization Engine operates as the primary access layer of the RAIDO platform and therefore interacts with virtually all platform components.

Orchestration is handled through Apache Airflow, which schedules and executes the preprocessing pipelines triggered from the interface. Through this interaction, users can configure, launch, and monitor preprocessing tasks while receiving status updates regarding execution progress. Federated Learning, Explainability, and Optimisation tasks are currently dispatched directly to their respective backend services, with ongoing work to route all task execution uniformly through the orchestration layer.

The component interacts directly with the Data Lake to retrieve and manage datasets, AI models, reports, and other artefacts. Users can browse, search, filter, edit, download,

and delete datasets and models through dedicated repository interfaces. Dataset and model uploads are performed within the corresponding task workflows rather than directly in the repositories, ensuring that every uploaded artefact is associated with a concrete processing task.

The Visualization Engine additionally provides user interfaces for the services implemented by the data preprocessing, federated learning, model distillation, data generation, and explainability components. Dedicated task configuration wizards allow users to execute image preprocessing, timeseries preprocessing, federated learning, explainability analysis, and optimization workflows without directly interacting with the underlying services.

For explainability and trustworthiness analysis, the component presents trust-level evaluations and explainability results through the platform's JSON and PDF report viewers. Optimization and benchmarking results follow the same retrieval and presentation mechanism. The component also interacts with authentication and authorization services through Keycloak and OAuth2-Proxy, ensuring secure access control, role-based permissions, and centralized session management throughout the platform. Finally, integration with a dedicated Green Monitoring User Interface (GMUI), providing users with monitoring information regarding resource utilization and energy consumption, is foreseen as part of the platform's evolution.

Through these interactions, the Visualization Engine acts as the unifying access layer connecting users with all major services and functionalities of the RAIDO ecosystem.

3.9.4 Additions, improvements, and final version of the component

While the original architecture focused primarily on adaptive visualizations and reporting functionalities, the final implementation evolved into a fully operational platform interface that supports the complete lifecycle of AI workflows. The development followed an iterative approach consisting of four major platform versions, progressively expanding functionality, backend integration, usability, and monitoring capabilities.

The first version established the visual structure and navigation model of the platform through a prototype implementation based on Vue.js and CoreUI components. The interface introduced the initial login page, user profile pages, repository views, task workflows, reporting dashboards, and basic visualization capabilities. Although the content was primarily static and backend integrations were not yet available, this version provided a common reference point for platform design and user interaction.

The second version transformed the prototype into a functional platform by introducing real backend connectivity, repository management capabilities, integration with the Data Lake, redesigned task creation workflows, and Keycloak-based authentication. This version established the first operational interaction between users and the underlying RAIDO services.

The third version focused on infrastructure improvements and operational deployment. The platform was migrated to Jetson-based edge infrastructure and integrated with Apache Airflow to support the execution of actual AI workflows. Image preprocessing became the first fully operational processing task available through the user interface, demonstrating the transition from a conceptual platform to an operational environment.

The fourth and final version delivered the complete production-ready RAIDO platform. The authentication architecture was redesigned around OAuth2-Proxy and Keycloak, providing centralized, server-side session management and a single, unified security boundary in front of all backend services. As a result, the frontend communicates with backend services using only session cookies, with no token-handling logic remaining in the application code. The architecture of this version is organized into three distinct layers, a security layer comprising the OAuth2-Proxy and Keycloak, the Vue.js-based frontend, and a backend cluster of services centered on the Data Lake as a unified storage layer, as illustrated in Figure 15.

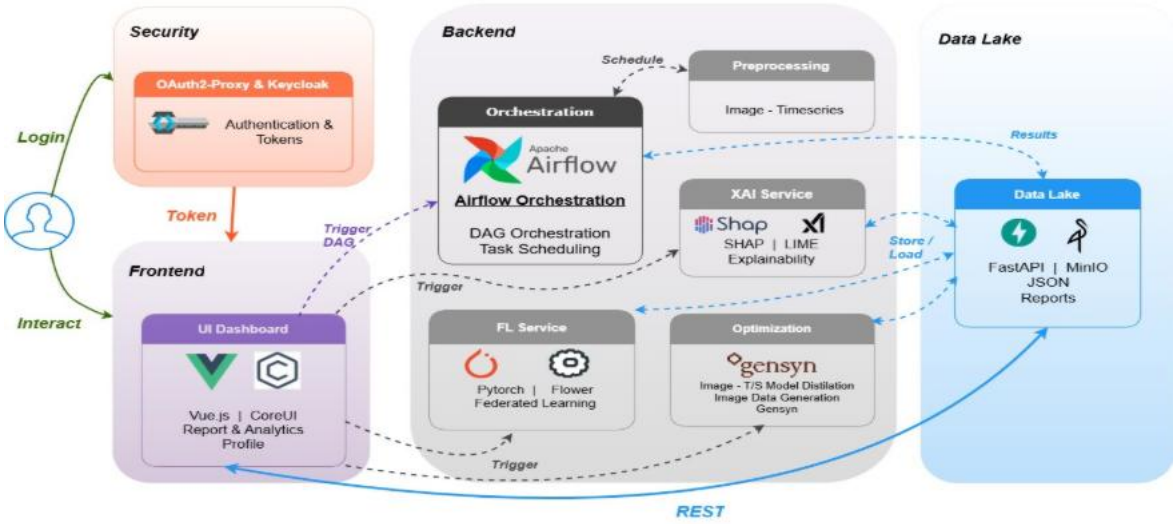

Figure 15: RAIDO platform v4 architecture.

The final platform provides five task types accessible through the Start a New Task wizard: Image Preprocessing, Timeseries Preprocessing, Federated Learning, Explainable AI, and Optimisation. Federated Learning covers image classification, timeseries classification, tabular classification, forecasting, and continual image classification subtasks; the latter, referred to as Continual Federated Learning (CFL), allows a model previously trained through the platform to be retrained in a federated manner without losing previously learned knowledge. Each subtask offers configurable model architecture, training strategy, differential privacy, and secure aggregation. The Optimization task groups together Image Model Distillation, Timeseries Model Distillation, Image Data Generation, and Timeseries Data Generation, the latter two producing synthetic image and timeseries data using generative AI techniques. Each task

is dispatched to its respective backend service, tracked by a unique run identifier, and its output stored in the Data Lake and retrieved as a JSON or PDF report once the pipeline finishes.

This version also delivers a more capable Reports and Analytics page, which presents a filterable table of all tasks belonging to the logged-in user, with time filters for the past week, month, year, or the full history. Each row reflects the real status of the task, running, failed, or succeeded, and selecting a succeeded task loads its report output directly on the page, displayed either through a structured JSON viewer or an embedded PDF viewer depending on the format produced by the backend.

On the visual and structural side, the final version introduces an animated background, a fully responsive layout that adapts to mobile and tablet screens with a collapsing sidebar, and a custom-built navigation sidebar. A new Accessibility Toolbar, embedded in the sidebar and reachable from any page, provides controls for adjusting text size in steps and switching visual modes including high contrast and negative contrast, with settings applied globally and persisted throughout the session. These additions directly address the accessibility requirements set out at the start of the project and align the interface with WCAG principles. The final platform therefore fulfills its role as the central interaction layer of the RAIDO ecosystem while supporting all major project requirements related to usability, accessibility, transparency, and Human-in-the-Loop interaction.

An important addition introduced during the final stages of development is the Green Monitoring User Interface (GMUI), which extends the Visualization Engine with advanced monitoring and sustainability-analysis capabilities. GMUI is specifically designed to provide task-level monitoring information regarding energy consumption, hardware utilization, and execution performance of Kubernetes-based workloads.

Upon completion of a task, users can inspect detailed monitoring metrics through interactive dashboards. The component receives user identifiers and task identifiers from the main RAIDO platform and retrieves monitoring information from Kepler, Prometheus, and InfluxDB services. Based on the collected information, GMUI generates a series of advanced visualizations and analytical metrics, including task duration, energy consumption, estimated carbon emissions using EU grid-intensity calculations, Gantt-style pod execution timelines, pod-level memory utilization breakdowns, runtime heatmaps, and resource-distribution radar charts.

To improve scalability and usability, the monitoring framework incorporates dynamic time-series resampling mechanisms and advanced filtering capabilities that enable users to investigate execution behavior at both task and pod level. Through these capabilities, GMUI complements the core Visualization Engine by providing detailed

observability, sustainability monitoring, and Green AI assessment functionalities, further strengthening the platform's support for energy-efficient and trustworthy AI deployment.

3.10 Self-evolving green-AI and data orchestration

3.10.1 Purpose and role within RAIDO

The Self-Evolving Green-AI and Data Orchestrator constitutes the central coordination mechanism of the RAIDO platform and is responsible for managing data processing, AI optimization, workflow execution, and resource-aware orchestration activities across the Edge-to-Cloud (E2C) continuum. Its primary objective is to autonomously construct, execute, monitor, and continuously optimize AI and data pipelines while satisfying user-defined constraints related to performance, energy consumption, resource utilization, and sustainability.

The component acts as the intelligence layer that connects end-users, datasets, AI models, optimization services, explainability mechanisms, and deployment infrastructures. Through Human-in-the-Loop (HITL) capabilities, users can define optimization objectives, upload datasets and models, configure workflows, and provide feedback that is subsequently incorporated into orchestration decisions. The component continuously evaluates alternative data transformation, training, optimization, and deployment strategies, selecting the most suitable workflow configurations according to the specified objectives.

Within the final RAIDO platform, the Green-AI Orchestrator serves as the backbone of all major optimization workflows, coordinating preprocessing, synthetic data generation, federated learning, model distillation, explainability analysis, and deployment operations across heterogeneous infrastructures while maintaining alignment with RAIDO's Green AI objectives.

3.10.2 Architecture and design principles

The initial architecture of the Green-AI Orchestrator was introduced in Deliverable D2.3 as a self-adaptive orchestration framework capable of coordinating AI and data workflows across distributed infrastructures. The original design focused on providing workflow orchestration, AI lifecycle management, autonomous decision making, and Human-in-the-Loop support through a modular and extensible architecture.

The design was based on several key principles. First, orchestration decisions should consider both performance and sustainability objectives rather than focusing solely on AI accuracy. Second, all optimization workflows should remain transparent and explainable through user interaction and monitoring mechanisms. Third, the orchestration framework should remain independent of specific AI models and support heterogeneous workflows across multiple domains and pilot environments.

The initial architecture envisioned seamless integration with the Data Lake, Adaptive Visualization Engine, Explainable AI services, Synthetic Data Generation, Federated Learning services, and Edge-to-Cloud deployment infrastructures. Communication between components was designed around APIs and metadata exchanges, allowing the orchestrator to coordinate distributed optimization workflows while maintaining modularity and scalability.

The high-level architecture of this component is shown in Figure 16.

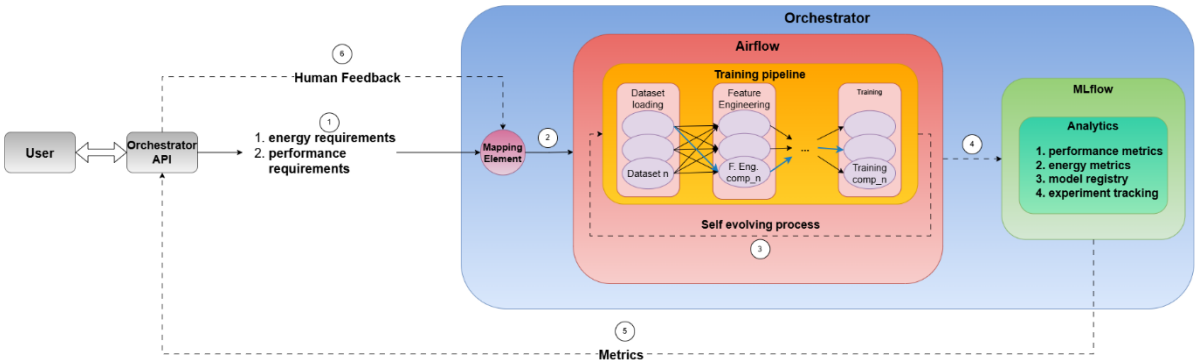

Figure 16. Green-AI component: Internal architecture.

3.10.3 Interactions with other components

The Green-AI Orchestrator interacts extensively with almost all core components of the RAIDO platform.

The component communicates directly with the Data Lake to retrieve datasets, AI models, metadata, and optimization artefacts required during workflow execution. It coordinates preprocessing operations with the Data Enrichment component and invokes the Synthetic Data Generation component whenever additional data augmentation or balancing is required.

The orchestrator also interacts with the FEDCL component to coordinate federated and continual learning workflows, while model distillation services are integrated to support the creation of lightweight and energy-efficient AI models. Explainability services are triggered through the XAI component to provide transparency and trustworthiness assessments for optimization outcomes.

Through the Visualization Engine, users interact with the orchestrator by defining workflow parameters, monitoring execution progress, reviewing optimization results, and providing Human-in-the-Loop feedback. The orchestrator additionally exchanges information with the Resource Manager to support execution planning across the Edge-to-Cloud infrastructure and with the blockchain services to guarantee traceability, auditability, and integrity of optimization activities.

3.10.4 Additions, improvements, and final version of the component

Since the publication of D2.3, the Green-AI Orchestrator evolved from a conceptual architecture into a fully operational orchestration framework integrated within the final RAIDO platform. The overall architecture and core logic proposed during the initial stages of the project remain valid and largely intact, with the final implementation realizing that vision through a set of modular, config-driven Airflow DAGs.

The final implementation is based on Apache Airflow, which was selected as the primary orchestration framework due to its workflow-as-code paradigm, scalability, extensibility, monitoring capabilities, and suitability for distributed Edge-to-Cloud environments. Through Directed Acyclic Graphs (DAGs), Airflow provides reproducible, transparent, and auditable workflow execution while supporting complex dependencies between tasks and services. Each DAG exposes a uniform REST-triggered interface, allowing the Orchestrator to compose and launch multi-step AI pipelines dynamically without requiring code changes per execution.

The implemented orchestration framework includes the Airflow Scheduler, Web Server, Metadata Database, Executors, and custom operators specifically developed for RAIDO services. These operators enable communication with the Data Lake, AI services, platform APIs, deployment infrastructures, and monitoring components.

Several operational workflows were implemented and validated, including image preprocessing, time-series preprocessing, synthetic data generation, Stable Diffusion image generation, Federated Continual Learning (FEDCL), model distillation, and Explainable AI workflows. Regardless of pipeline type, all DAGs follow a consistent three-stage execution pattern: triggering the downstream service via its REST API and registering the task in the Data Lake, continuously polling for task completion or failure, and automatically notifying the user by email upon reaching a terminal status. Each workflow can be executed independently or combined as part of larger optimization pipelines according to user-defined requirements.

A feedback-driven optimization mechanism was additionally incorporated to support self-evolving behavior. The Orchestrator communicates with end-users exclusively through the Adaptive Visualization Engine (AVE), which compiles user inputs into a structured JSON payload forwarded to the Airflow REST API, keeping the execution layer fully decoupled from the frontend. Two deviations from the initial design are noted in this area: the bidirectional HITL signaling pathway is realized through user-initiated re-runs via the platform dashboard rather than an automated feedback channel, preserving human oversight while deferring full automation to a future iteration; and energy efficiency metrics are consolidated within the Feedback Loop component (C08) rather than tracked independently per DAG, ensuring a single consistent source of energy metrics across the platform.

The final implementation was deployed and validated across multiple execution environments, including local development infrastructures, virtualized environments, NVIDIA Jetson edge devices, and Kubernetes clusters. This multi-environment deployment demonstrates the ability of the orchestrator to manage AI workflows across heterogeneous Edge-to-Cloud infrastructures while maintaining portability, scalability, and operational consistency.

As a result, the Green-AI Orchestrator represents the operational core of the final RAIDO platform, enabling autonomous, explainable, sustainable, and adaptive AI lifecycle management.

3.11 Networking Management and AI-based E2C Continuum Resource Allocation & Deployment (Resource Manager)

3.11.1 Purpose and role within RAIDO

The Resource Manager is responsible for orchestrating the execution and deployment of AI workloads across the Edge-to-Cloud continuum while ensuring efficient utilization of available computational, storage, and networking resources. Its primary objective is to determine where and how AI tasks should be executed in order to satisfy Service Level Objectives (SLOs) related to latency, energy consumption, throughput, and resource efficiency.

The component continuously evaluates the state of the underlying infrastructure and dynamically allocates workloads across edge and cloud environments. Through AI-assisted scheduling mechanisms, the Resource Manager supports intelligent placement decisions that maximize performance while minimizing operational costs and energy consumption.

In addition to computational resources, the component considers networking conditions and infrastructure-level monitoring information to improve deployment decisions. Consequently, it acts as the operational execution layer that transforms orchestration decisions into actual deployments across the Edge-to-Cloud infrastructure.

3.11.2 Architecture and design principles

The initial architecture introduced in D2.3 combined resource management and networking management functionalities under a common Edge-to-Cloud deployment framework.

The design envisioned a resource-aware execution environment capable of monitoring infrastructure resources and dynamically scheduling workloads according to application requirements. Resource allocation decisions were designed to consider energy consumption, computational capacity, memory utilization, bandwidth availability, latency constraints, and workload characteristics.

Networking management capabilities were additionally incorporated to support virtual overlay networks, cloud-edge connectivity, cluster management, and the collection of network metrics that could contribute to placement and scheduling decisions. The architecture also envisioned integration with Kubernetes infrastructures and Software Defined Networking (SDN) technologies to facilitate deployment flexibility and scalability.

Overall, the component was designed to support intelligent workload placement across heterogeneous infrastructures while maintaining efficient resource utilization and operational sustainability.

The high-level architecture of this component is shown in Figure 17.

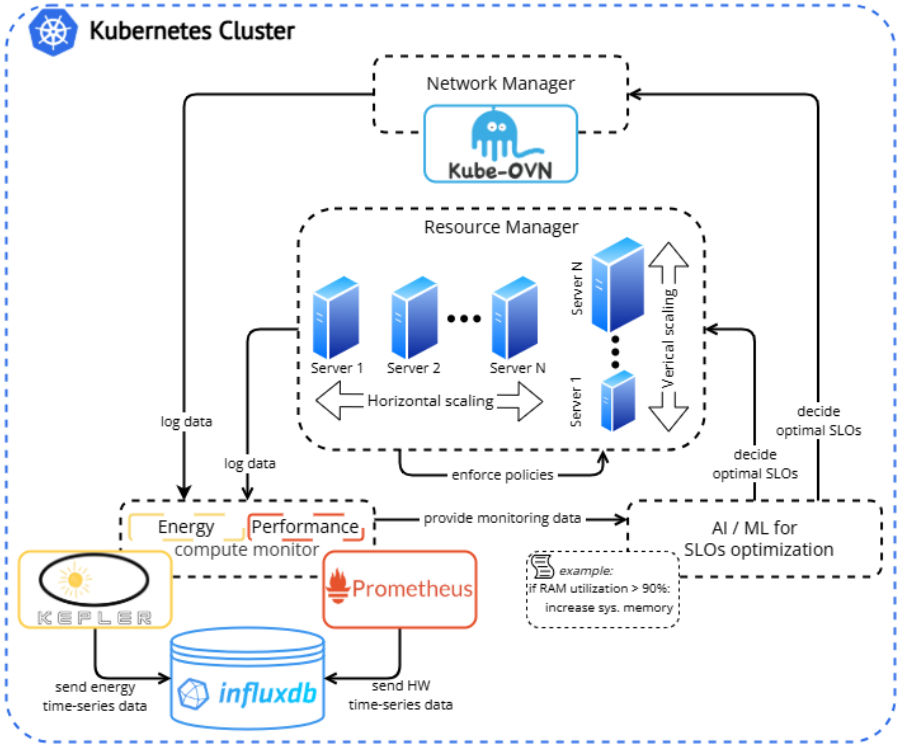

Figure 17. Resource Manager component: Internal architecture.

3.11.3 Interactions with other components

The Resource Manager interacts directly with the Green-AI Orchestrator, receiving workflow execution requirements and deployment requests generated during optimization activities.

The component exchanges monitoring information with the Data Lake and receives infrastructure metrics related to computational resources, workload execution, system performance, and resource utilization. These metrics are subsequently used to support placement decisions and deployment optimization.

The Visualization Engine receives monitoring information from the Resource Manager and presents resource utilization, deployment status, energy consumption metrics,

execution performance indicators, and infrastructure analytics through dedicated dashboards.

The Resource Manager additionally utilizes networking information, including bandwidth availability, latency measurements, and communication performance metrics, to support placement decisions across distributed environments. These interactions ensure that deployment decisions consider both computational and networking characteristics of the underlying infrastructure.

3.11.4 Additions, improvements, and final version of the component

Since D2.3, the Resource Manager evolved significantly through the implementation activities performed within Task 5.3.

The final implementation focuses on AI-assisted Edge-to-Cloud deployment and resource management across heterogeneous infrastructures. The component supports deployment on both NVIDIA Jetson edge devices and Kubernetes-managed execution environments, enabling flexible workload execution across distributed computing resources.

A major enhancement introduced during implementation is the integration of infrastructure monitoring services capable of collecting detailed information regarding resource utilization, hardware consumption, task execution behavior, and energy usage. These monitoring capabilities provide the foundation for intelligent scheduling decisions and Green AI optimization strategies.

The deployment framework was validated on NVIDIA Jetson devices, demonstrating support for edge-level AI processing and lightweight execution scenarios. Subsequently, the deployment environment was extended towards Kubernetes-based infrastructures, allowing containerized services and AI workloads to be executed in a scalable and distributed manner.

Additional monitoring capabilities were incorporated through integration with Prometheus, InfluxDB, and Kepler-based monitoring services. These services collect metrics related to CPU utilization, memory consumption, workload execution, hardware utilization, energy consumption, and infrastructure behavior. The collected information is subsequently visualized through dedicated monitoring dashboards and used to support optimization decisions.

The component also supports the generation of execution-level metrics and sustainability indicators, including energy consumption measurements and hardware utilization analytics. These metrics contribute directly to RAIDO's Green AI objectives by enabling detailed assessment of resource efficiency and workload performance across the Edge-to-Cloud continuum.

The final implementation therefore transforms the Resource Manager into an operational deployment and monitoring framework capable of supporting intelligent workload execution, infrastructure optimization, sustainability monitoring, and resource-aware AI deployment throughout the RAIDO platform.

3.12 Blockchain & Smart Contracts with IPFS Support

3.12.1 Purpose and role within RAIDO

The Blockchain & Smart Contracts with IPFS Support component provides the trust, traceability and transparency layer of the RAIDO platform. Its primary objective is to establish a tamper-evident monitoring infrastructure capable of recording, validating and auditing key events occurring throughout the AI and data optimization lifecycle.

The component enables transparent process monitoring by combining permissioned blockchain technologies with decentralized off-chain storage mechanisms. Rather than storing datasets, models, or optimization artefacts directly on-chain, the component records immutable evidence regarding lifecycle transitions, validation outcomes, optimization results, ownership information and certification-relevant events. This approach ensures accountability, reproducibility, auditability and regulatory compliance while maintaining low operational overhead.

Within RAIDO, the component serves as a trusted evidence layer supporting data enrichment, synthetic data generation, AI model optimization, federated learning, deployment, explainability and benchmarking activities. Through smart contracts and distributed storage technologies, the component provides verifiable records of process execution while preserving scalability and privacy requirements.

3.12.2 Architecture and design principles

The initial architecture introduced in D2.3 envisioned a blockchain-enabled certification framework integrating smart contracts, decentralized storage mechanisms and organizational governance procedures to support AI model optimization and monitoring activities.

The design was based on several core principles. First, the blockchain infrastructure should act as a trust anchor rather than a primary data repository. Second, large artefacts and datasets should remain off-chain while preserving cryptographic linkage to blockchain records. Third, all lifecycle transitions should be governed through programmable smart contracts capable of enforcing validation and certification rules. Finally, the infrastructure should support multi-organizational governance, allowing multiple stakeholders to participate in validation and monitoring activities.

The original architecture therefore combined Hyperledger Fabric for permissioned distributed ledger management with IPFS for decentralized storage. Smart contracts

were expected to manage lifecycle states, ownership information, validation procedures and certification activities, while external services would interact with the blockchain through dedicated APIs.

The architecture also adopted an ERC-721-inspired token model, not for financial purposes, but as a mechanism for uniquely identifying AI models, datasets, optimization runs and metadata artefacts throughout their lifecycle. This approach established a consistent framework for provenance tracking, version control and auditability across the RAIDO ecosystem.

The high-level architecture of this component is shown in Figure 18.

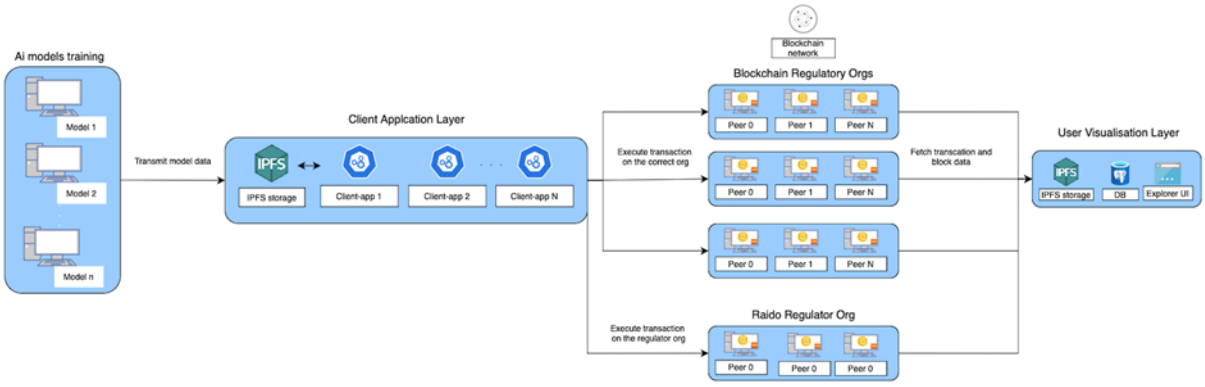

Figure 18. Blockchain component: Internal architecture.

3.12.3 Interactions with other components

The Blockchain component interacts with several core components of the RAIDO platform.

Blockchain events are triggered by ORCH during workflow execution, while references to datasets, models, execution metadata, and optimization artefacts stored in the DATALAKE are certified through blockchain transactions and IPFS content identifiers.

The Data Lake stores datasets, metadata and AI artefacts that may be referenced through blockchain records. Rather than storing the actual content on-chain, the component stores content identifiers (CIDs) generated through IPFS, creating verifiable links between blockchain transactions and stored artefacts.

The Green-AI Orchestrator can trigger monitoring events whenever workflow stages are completed, validated or deployed. This enables the blockchain infrastructure to capture lifecycle transitions throughout the optimization process.

The Visualization Engine retrieves certification information, transaction histories, lifecycle states and audit records through the Blockchain APIs and presents them to end users through dedicated dashboards and monitoring interfaces.

The component additionally supports future integration with pilot environments, explainability services, benchmarking components and deployment infrastructures by providing a common trust and traceability framework for all optimization activities.

3.12.4 Additions, improvements, and final version of the component

Since D2.3, the Blockchain component evolved from a conceptual certification framework into a fully implemented monitoring and trust infrastructure based on Hyperledger Fabric and IPFS, with end-to-end certification automation delivered through Apache Airflow DAG integration. The final implementation is described in detail in Deliverable D5.3 and provides a complete proof-of-concept deployment supporting transparent process monitoring across the RAIDO platform.

The final architecture adopts a consortium-style permissioned blockchain model implemented using Hyperledger Fabric. This approach was selected because it supports controlled participation, organization-aware governance, identity management and efficient operation within multi-organizational environments. The blockchain network utilizes Membership Service Providers (MSPs), Certificate Authorities (CAs), endorsing peers and ordering services to provide secure transaction validation and lifecycle management.

A major enhancement introduced during implementation is the adoption of a lightweight monitoring architecture in which the blockchain records only certification-relevant information, including lifecycle transitions, timestamps, validation outcomes, actors, ownership information and references to external artefacts. Detailed outputs, datasets, optimization logs, benchmarking reports and metadata are first persisted to MinIO and subsequently stored off-chain using the InterPlanetary File System (IPFS), while their corresponding Content Identifiers (CIDs) are anchored on-chain to guarantee integrity and traceability. The Pinata service is utilized to ensure persistent availability of IPFS artefacts.

The final implementation introduces an ERC-721-inspired token model implemented as Node.js chaincode that provides unique lifecycle-aware identifiers for AI models, datasets, optimization runs and metadata artefacts. Each token is uniquely identified by its Airflow dag_run_id and progresses through four strictly enforced lifecycle states: INITIALIZED → IN_PROGRESS → COMPLETED / FAILED. Unlike traditional NFTs, these tokens are not intended for financial transactions but instead serve as digital identities supporting provenance tracking, ownership management, version control and lifecycle traceability. Each token maintains a per-token lifecycleHistory array capturing actor identity, transaction timestamp, the IPFS CID of the associated artefact, ownership information and validation status throughout its lifecycle.

The smart contract layer was implemented using Hyperledger Fabric chaincode in Node.js and includes RAIDO-specific lifecycle management functions. Key functions

include RecordBaseline, which stores pre-optimization metric values on-chain after initial pipeline execution, and RecordFinalResult, which extracts post-optimization values and compares them against the stored baseline using a built-in metric direction map that distinguishes higher-is-better metrics (such as accuracy and R^2) from lower-is-better metrics (such as loss, MAE, FID, duration and energy), transitioning the token to COMPLETED if no regressions are detected or FAILED otherwise. The contract natively supports both Optimisation and Federated Learning output formats, ensuring all major RAIDO pipeline types are certified through the same on-chain logic. Through endorsement policies, critical lifecycle transitions can require validation from supervisory organizations, enabling organization-aware governance and certification workflows.

To facilitate integration with the rest of the RAIDO platform, a dedicated client application (raido_client_app) and REST API layer was developed using Express.js and the Hyperledger Fabric Gateway SDK, handling identity, channel targeting and chaincode invocation. This layer enables orchestration services, dashboards and external applications to interact with the blockchain infrastructure without requiring direct blockchain knowledge. The end-to-end certification flow is fully automated via a submit_to_blockchain task embedded in both the optimization and federated_learning DAGs, which upon pipeline completion executes five sequential steps: fetch the output JSON from MinIO, pin it to IPFS via Pinata, mint the NFT on-chain, invoke RecordBaseline and invoke RecordFinalResult. More broadly, all RAIDO Airflow DAGs — including those wrapping the XAI engine and preprocessing pipelines — share a common notification architecture whereby each DAG automatically dispatches an email notification to the user via the Data Lake notification service upon reaching a terminal state, ensuring consistent and timely updates across every external component integrated into the platform.

The final deployment architecture follows Hyperledger Fabric's execute-order-validate transaction model. Transactions are first simulated by endorsing peers, validated according to endorsement policies, ordered through a Raft-based ordering service and finally committed to the distributed ledger. This approach provides efficient governance, scalability and secure lifecycle management while ensuring that all recorded events remain immutable and auditable.

An additional enhancement introduced during D5.3 is the incorporation of operational monitoring and performance evaluation mechanisms. Hyperledger Caliper was used to benchmark representative blockchain operations, including token creation, baseline recording, result submission, lifecycle queries and deletion operations. Runtime monitoring capabilities were also implemented using Prometheus and Grafana dashboards, enabling continuous observation of peers, orderers, chaincode execution,

resource utilization and transaction performance. These mechanisms provide operational visibility and support future scalability assessments.

Overall, the final Blockchain & Smart Contracts with IPFS component establishes a secure, auditable and organization-aware trust infrastructure for the RAIDO platform. By combining Hyperledger Fabric, smart contracts, decentralized storage, Apache Airflow-driven certification automation, MinIO-based result persistence, REST-based integration services and operational monitoring mechanisms, the component provides transparent process monitoring, certification support, lifecycle traceability and immutable evidence management throughout the AI and data optimization lifecycle.

This is actually one of the strongest components in D2.4 because, unlike Network Management, there is a complete implemented architecture, technology stack, deployment, APIs, smart contracts, monitoring infrastructure, benchmarking results and operational validation coming directly from D5.3.

4 PLATFORM DEPLOYMENT AND INTEGRATION

The previous chapters presented the final RAIDO platform architecture together with the technological components that constitute the platform and their role within the overall ecosystem. Building upon this architectural foundation, the present chapter focuses on the operational realization of the platform, describing how the individual services were deployed, integrated, and brought together into a unified operational environment.

The deployment and integration activities performed throughout the project transformed the initial architectural design into a fully operational platform capable of supporting data-centric and model-centric optimization processes, workflow orchestration, deployment management, monitoring, explainability, and governance services. Through the successful integration of the technological outcomes developed across Work Packages 3, 4, and 5, the RAIDO platform demonstrates the coordinated operation of its individual services within a common execution environment, supporting reliable, trustworthy, explainable, and sustainable AI workflows. The following sections present the final deployment architecture, summarize the achieved platform integration status, and discuss the mechanisms supporting secure, scalable, and sustainable platform operation.

4.1 Final deployment architecture

The final deployment architecture realizes the modular design of the RAIDO platform by deploying its technological components as independent services within a distributed execution environment. This deployment strategy enables the platform to maintain loose coupling between services while supporting independent deployment, maintenance, scalability, and operational flexibility. By separating deployment concerns from the functional architecture, the platform can efficiently execute complex AI workflows across heterogeneous computational infrastructures.

The deployment model supports cloud, edge, and hybrid computing environments, allowing platform services to be executed on the most appropriate computational infrastructure according to application requirements, data locality, latency constraints, privacy considerations, and resource availability. This flexibility is particularly important given the heterogeneous nature of the RAIDO pilot environments, ranging from smart energy systems and smart farming infrastructures to healthcare and industrial robotics applications.

The individual platform services are deployed independently and may operate either as standalone services or as part of coordinated AI optimization workflows. Workflow execution is coordinated through the orchestration mechanisms of the platform, allowing optimization, trustworthiness, monitoring, and governance services to be dynamically combined according to the requirements of each application scenario. This

deployment approach enables individual services to evolve independently while preserving interoperability across the overall platform ecosystem.

Persistent platform resources, including datasets, AI models, metadata, monitoring information, benchmarking results, deployment artefacts, and optimization outputs, are managed through the DATALAKE. Access to these resources is provided through the backend services and APIs, enabling secure and controlled information exchange between the deployed platform services while ensuring consistency across the distributed deployment environment.

The deployment architecture is completed by the Visualization Engine (UI), which provides users with a unified access point to the platform services, together with the resource management and networking services responsible for workload scheduling, deployment management, resource allocation, and communication across the distributed infrastructure. These services collectively support the reliable execution of AI workflows while enabling efficient utilization of the available computational resources.

Overall, the final deployment architecture demonstrates the successful realization of the RAIDO platform as an integrated, distributed, and operational AI ecosystem. The combination of modular service deployment, centralized data management, orchestration, resource management, and standardized communication mechanisms enables scalable and interoperable AI workflows while supporting the diverse operational requirements of the RAIDO application domains.

4.2 Platform integration status

The integration of the individual technological components into a unified and operational platform constituted one of the principal objectives of the RAIDO project. While Chapters 2 and 3 presented the final architecture and the individual platform services, the integration activities focused on ensuring that these independently developed components could operate together as a coherent and interoperable ecosystem.

The integration process involved the establishment of standardized communication interfaces, backend services, workflow orchestration mechanisms, shared data management, deployment procedures, authentication services, monitoring capabilities, and governance mechanisms. Rather than functioning as isolated services, the platform components were successfully interconnected through the common architectural infrastructure, enabling the coordinated execution of end-to-end AI optimization workflows.

The successful completion of the integration activities demonstrated the interoperability of the platform across the different technological domains addressed within the project. The AI optimization services, trustworthiness components, deployment infrastructure, and supporting backend services operate within a common execution environment,

allowing data, models, metadata, monitoring information, and optimization artefacts to be exchanged seamlessly throughout the platform.

The integrated platform was subsequently validated through the RAIDO pilot activities, demonstrating the coordinated operation of the technological components across diverse application domains and heterogeneous computational environments. These validation activities confirmed that the final platform successfully supports the complete execution of AI workflows, from data preparation and model optimization to deployment, explainability, monitoring, and governance, thereby achieving the integration objectives established for the project.

4.3 Security, governance, and access control

Security, governance, and controlled access constitute fundamental operational principles of the RAIDO platform. Given the heterogeneous nature of the supported application domains and the potentially sensitive nature of the processed datasets and AI models, the platform incorporates a comprehensive security framework that protects platform resources while ensuring secure communication, controlled access, transparency, and trustworthy operation across the distributed deployment environment.

Access to the platform is managed through the Backend Services and API Layer, which provides centralized authentication and authorization services for all platform interactions. The backend implements an OAuth2-based authentication framework integrated with Keycloak for identity and access management, while Role-Based Access Control (RBAC) policies ensure that users can only access the datasets, services, and platform functionalities associated with their assigned roles and responsibilities. Authenticated sessions are maintained using JSON Web Tokens (JWT), enabling secure, stateless communication between the Visualization Engine, backend services, and the individual platform components.

Secure communication between the platform services is further supported through the standardized REST APIs exposed by the Backend Services and API Layer. These APIs provide controlled access to the platform resources while abstracting the implementation details of the individual services. All interactions with datasets, AI models, metadata, monitoring information, benchmarking results, and deployment artefacts are mediated through the backend services, ensuring consistent enforcement of authentication, authorization, and access control policies across the platform.

Governance is reinforced through the centralized management of platform resources within the DATALAKE, which supports controlled access, versioning, lifecycle management, metadata management, and information traceability. This centralized governance model improves the reproducibility of AI workflows, facilitates the

management of optimization artefacts throughout their lifecycle, and ensures consistent handling of the information exchanged between the platform services.

Operational transparency and accountability are further strengthened through the platform's trustworthiness services. Explainability mechanisms enable users to understand the behavior and predictions of AI models, benchmarking services continuously evaluate optimization outcomes and system performance, while blockchain-enabled provenance management maintains immutable records of critical workflow events and execution metadata. Together, these capabilities enhance transparency, traceability, auditability, and trust throughout the execution of AI workflows.

The combination of centralized authentication, role-based authorization, secure API communication, controlled data management, provenance tracking, explainability, and continuous performance evaluation establishes a comprehensive security and governance framework that supports the reliable operation of the RAIDO platform across heterogeneous deployment environments. Detailed implementation aspects related to backend security, API implementation, infrastructure protection, authentication services, and data management are presented in Deliverables D3.3 and D6.1 and are therefore not repeated in the present document.

4.4 Scalability, extensibility, and future evolution

The final RAIDO platform has been designed not only to satisfy the requirements of the project use cases but also to provide a sustainable technological foundation capable of supporting future research, industrial adoption, and the continuous evolution of Artificial Intelligence technologies. The architectural and deployment decisions adopted throughout the project prioritize modularity, interoperability, flexibility, and operational scalability, enabling the platform to evolve without requiring fundamental modifications to its core structure.

Scalability is achieved through the combination of the platform's modular service-oriented architecture and its distributed deployment model. Individual platform services can be deployed, maintained, updated, and scaled independently according to operational requirements, allowing computational resources to be allocated dynamically across cloud, edge, and hybrid computing environments. Resource provisioning and workload management are coordinated by RSCMANAGER, which continuously evaluates infrastructure availability and application requirements to optimize service deployment. Through dynamic workload scheduling and support for both horizontal scaling, by increasing the number of service instances, and vertical scaling, by adjusting the computational resources allocated to existing services, the platform efficiently accommodates varying workload demands while maintaining service availability and resource utilization.

Extensibility is further supported through the clear separation of architectural responsibilities and the adoption of standardized REST APIs exposed through the Backend Services and API Layer. This approach enables new AI optimization techniques, explainability services, trustworthiness mechanisms, analytical capabilities, and domain-specific components to be incorporated into the platform without affecting the operation of the existing services. Similarly, the standardized communication interfaces facilitate the integration of external services and future technological developments while preserving interoperability across the overall ecosystem.

The platform architecture is equally adaptable to future application domains. Although the final implementation has been validated through the RAIDO pilot environments in smart energy, smart farming, healthcare, and industrial robotics, the underlying architectural principles remain domain-independent. Consequently, the same technological framework can support additional AI-driven application scenarios through the integration of new datasets, AI models, optimization services, and deployment infrastructures while reusing the existing platform capabilities.

Overall, the final RAIDO platform establishes a scalable, extensible, and sustainable AI ecosystem capable of supporting the continuous evolution of AI technologies beyond the lifetime of the project. Through its modular architecture, standardized integration mechanisms, intelligent resource management, and distributed deployment capabilities, the platform provides a robust foundation for future research, industrial exploitation, and the adoption of reliable, transparent, and sustainable AI solutions across diverse operational environments.

5 REQUIREMENTS TRACEABILITY AND ARCHITECTURAL VALIDATION

The development of the RAIDO platform was guided by the stakeholder requirements, system requirements, and architectural objectives identified during the early phases of the project and documented in D2.1 and D2.3. These requirements established the foundation for the architectural design of the platform and guided the implementation, integration, deployment, and validation activities performed throughout the project lifecycle.

While D2.3 introduced the initial Requirements Traceability Matrix (RTM) and established the relationships between stakeholder requirements, non-functional requirements, actors, and architectural components, the purpose of this chapter is to demonstrate how these requirements have been addressed by the final RAIDO platform architecture and validated through the implementation and integration activities performed across Work Packages 3, 4, 5, and 6.

The final platform integrates all core architectural components into a unified ecosystem supporting data management, AI model optimization, explainability, orchestration, deployment, monitoring, benchmarking, resource management, and governance functionalities. The successful deployment and validation activities documented in D6.1 provide evidence that the implemented platform satisfies the functional and non-functional objectives identified during the requirements analysis phase.

5.1 Architectural validation

The validation of the RAIDO architecture was performed through a progressive process involving component implementation, service integration, platform deployment, and pilot execution activities. Rather than validating individual services in isolation, the project focused on demonstrating the operational behavior of the complete ecosystem and assessing the ability of the platform components to cooperate within integrated optimization workflows.

The validation activities confirmed the successful interoperability of the platform services and demonstrated the capability of the architecture to support the complete AI lifecycle. Data preparation, enrichment, synthetic data generation, model optimization, federated and continual learning, explainability assessment, benchmarking, deployment, monitoring, and governance workflows were successfully integrated within a common platform environment and executed through the orchestration mechanisms provided by ORCH.

The deployment and evaluation activities performed within the RAIDO pilots further demonstrated the applicability of the architecture across heterogeneous application domains and infrastructure environments. The successful integration of all major platform components confirms the maturity of the developed ecosystem and validates its ability to support end-to-end AI workflows spanning data management, optimization, deployment, monitoring, and continuous improvement activities. Detailed information regarding deployment activities, pilot execution, testing procedures, and evaluation results is provided in D6.1 and is therefore not repeated within this deliverable.

The validation outcomes indicate that the final architecture successfully supports the objectives of the project regarding interoperability, scalability, reliability, trustworthiness, explainability, sustainability, security, and operational efficiency. Furthermore, the successful integration of all major platform components demonstrates the maturity of the developed ecosystem and confirms its ability to support the complete Artificial Intelligence lifecycle across heterogeneous cloud, edge, and hybrid infrastructures.

5.2 Updated requirements traceability matrix

The Requirements Traceability Matrix (RTM) initially introduced in Deliverable D2.3 has been updated to reflect the final implementation status of the RAIDO platform and the completion of the integration, deployment, and validation activities performed throughout the project. The updated RTM provides traceability between the stakeholder requirements (StRs), non-functional requirements (NFRs), functional requirements (FRs), technical actors, and architectural components, demonstrating the alignment of the final platform with the objectives and specifications identified during the requirements analysis phase.

Beyond serving as a traceability mechanism, the RTM also acts as a verification instrument for the final platform architecture. It provides evidence that the identified requirements have been addressed through the implemented platform services and successfully integrated within the final RAIDO ecosystem. In this way, the matrix demonstrates how the individual architectural components collectively contribute to the realization of the platform capabilities and support the overall objectives of the project.

The RTM builds upon the requirements and specifications identified in D2.1 “Use Case, KPIs, Requirements, Specification, Slices & Technology Enablers Definition Report” and the initial architectural mappings established in Deliverable D2.3. The updated version reflects the final architecture presented in this deliverable and incorporates the outcomes of the implementation, deployment, integration, and validation activities performed across Work Packages 3, 4, 5, and 6.

Table 4 presents the updated Functional Requirements Traceability Matrix and establishes the relationships between the Functional Requirements (FRs), the

corresponding stakeholder requirements, non-functional requirements, technical actors, and final architectural components responsible for their realization. In addition, the matrix records the implementation and validation status of each requirement and provides implementation evidence demonstrating how the corresponding functionality has been realized within the final platform architecture. The table therefore provides direct traceability from the initial project requirements to the implemented platform services and validates the successful realization of the functional capabilities envisioned during the architectural design phase.

While Table 4 focuses on the realization of the functional requirements, Table 5 provides a complementary assessment of the non-functional requirements identified during the requirements analysis phase. The table evaluates the degree to which the final platform satisfies the quality attributes defined for the RAIDO ecosystem, including functional completeness, correctness, interoperability, operability, security, availability, adaptability, sustainability, modularity, and trustworthiness. For each non-functional requirement, the table provides validation evidence derived from the implemented architecture, platform services, deployment activities, and pilot evaluations.

Together, Tables 4 and 5 provide comprehensive evidence that the final RAIDO platform successfully satisfies both the functional and non-functional requirements established throughout the project. The traceability links maintained between stakeholder requirements, architectural components, technical actors, and validation outcomes demonstrate the consistency of the architectural design process and confirm that the implemented platform remains aligned with the original project vision and objectives.

The updated RTM demonstrates that the final architecture successfully satisfies the requirements identified throughout the project and provides the technological foundation necessary to support reliable, trustworthy, explainable, interoperable, secure, and sustainable AI workflows. The successful implementation, integration, deployment, and validation of the platform components confirm the maturity of the developed ecosystem and establish the final RAIDO platform as a coherent and operational framework capable of supporting the complete Artificial Intelligence lifecycle across heterogeneous application domains.

Table 3: Updated functional requirements traceability and validation matrix.

FR ID	Title	Satisifed?	Implementation / / Validation evidence	Link to StRs	Link to RAIDO NFRs	RAIDO actors	Link to component id
FR01	Scalable synthetic data generation for AI models	Yes	Addressed through DATAGEN, which provides synthetic data generation capabilities for image and time-series data using generative AI models and graphics-based simulation environments. The component is integrated with ORCH and DATALAKE, enabling generated synthetic datasets and related metadata to be used by downstream AI workflows.	R01, R27	SP1 - Functional completeness, SP3 - Time behavior, SP4 - Resource utilization, SP8 - Availability, SP14 - Adaptability	AE, DE	C03/DATAGEN
FR02	Enrichment and fusion of multimodal data	Yes	Addressed through ENRICH and DATALAKE. ENRICH supports data enrichment, curation, preprocessing, balancing, distillation, and federated mining activities, while DATALAKE provides the central information management layer for structured, semi-structured, unstructured, image, model, and time-series data using the final polyglot storage infrastructure.	R02	SP1 - Functional completeness, SP4 - Resource utilization, SP5 - Interoperability, SP8 - Availability, SP9 - Confidentiality	DP, AE, DE	C02/ENRICH, C06/DATALAKE
FR03	Federated and continual learning for scalable data processing	Yes	Addressed primarily through FEDCL, which provides Flower-based federated learning workflows, Federated Continual Learning strategies, and an FSNet-based continual learning workflow for time-series applications. The requirement is further supported by DISTIL, which provides energy-efficient knowledge distillation workflows for vision and time-series models. Together, these components support privacy-preserving distributed learning, continual adaptation, reduced data movement, and energy-efficient model optimization.	R03, R12, R13	SP2 - Functional correctness, SP3 - Time behavior, SP4 - Resource utilization, SP9 - Confidentiality, SP10 - Integrity	AE, DE, PA	C05/FEDCL, C04/DISTIL
FR04	End-to-end secure data management	Yes	Addressed through DATALAKE, UI, and BLC. DATALAKE provides secure storage and management of datasets, models, metadata, monitoring information, and artefacts. UI supports controlled user access to platform functionalities. BLC provides blockchain-based monitoring, traceability, provenance management, and immutable records of relevant platform activities, supporting auditability and governance.	R04	SP8 - Availability, SP9 - Confidentiality, SP10 - Integrity, SP11 - Authenticity, SP13 - Testability	AE, DE, PA	C06/DATALAKE, C09/UI, C12/BLC

FR ID	Title	Satisifed?	Implementation / / Validation evidence	Link to StRs	Link to RAIDO NFRs	RAIDO actors	Link to component id
FR05	Adaptive optimization for AI model training	Yes	Addressed through RLFEED, ORCH, and RSCMANAGER. RLFEED provides benchmarking and feedback-driven optimization based on performance, duration, and energy-related indicators. ORCH coordinates the execution of optimization workflows and service dependencies. RSCMANAGER supports resource allocation, workload scheduling, deployment management, and monitoring across cloud, edge, and hybrid infrastructures, including resource and energy-awareness capabilities.	R05, R06, R10, R11, R15, R16, R17, R18, R19, R21, R22, R26, R29, R30, R32, R34, R35	SP3 - Time behavior, SP4 - Resource utilization, SP7 - User engagement, SP8 - Availability, SP14 - Adaptability	AE, DE, PA	C08/RLFEED, C10/ORCH, C11/RSCMNGR
FR06	Compliance with regulatory and legal frameworks	Partially	Addressed primarily through BLC, which provides blockchain-based traceability, provenance management, immutable lifecycle records, and certification-oriented monitoring. Together with the platform authentication and governance mechanisms, these capabilities support auditability and accountability. However, compliance with specific regulatory and legal frameworks ultimately depends on the deployment context, applicable legislation, and organizational governance procedures, which extend beyond the technical scope of the RAIDO platform. Therefore, this requirement is considered partially satisfied.	R07, R20	SP1 - Functional completeness, SP9 - Confidentiality, SP10 - Integrity, SP12 - Modularity, SP13 - Testability	AE, PA	C12/BLC
FR07	Intuitive interface with HITL integration	Yes	Addressed through UI and XAI. UI provides the main user-facing environment for dataset management, workflow configuration, task execution, monitoring, reporting, and result inspection. XAI provides explainability, fairness assessment, trustworthiness evaluation, and interpretable outputs that support Human-in-the-Loop review and decision-making.	R08, R24, R25, R28	SP6 - Operability, SP7 - User engagement, SP2 - Functional correctness, SP12 - Modularity, SP14 - Adaptability	AE, PA	C07/XAI, C09/UI
FR08	Seamless integration with legacy systems	Partially	The final architecture provides standard REST APIs, modular workflows, and interoperable data management services that facilitate integration with external datasets, AI models, and third-party infrastructures. These mechanisms were successfully demonstrated within the RAIDO pilot environments. Nevertheless, complete integration with all legacy systems cannot be guaranteed, as it depends on the interfaces, constraints, and	R09, R33	SP1 - Functional completeness, SP5 - Interoperability, SP6 - Operability, SP12 - Modularity, SP14 - Adaptability	AE, DE, PA	C09/UI, C06/DATALAKE, C10/ORCH

FR ID	Title	Satisifed?	Implementation / / Validation evidence	Link to StRs	Link to RAIDO NFRs	RAIDO actors	Link to component id
			technologies of each external environment. Consequently, this requirement is considered partially satisfied.				

Table 4: Validation of RAIDO non-functional requirements.

NFR ID	Title	Satisifed?	Validation evidence
SP1	Functional completeness	Yes	The final platform integrates all major capabilities envisioned in D2.3, including data management, synthetic data generation, AI optimization, federated learning, explainability, orchestration, deployment, monitoring, benchmarking, and governance services.
SP2	Functional correctness	Yes	The integrated workflows were successfully executed and validated within the pilot environments described in D6.1, demonstrating the correct operation of the implemented services and optimization workflows.
SP3	Time behavior	Partially	Most workflows perform adequately, but execution time still depends on workload, infrastructure, AI model complexity, and distributed deployment. No strict timing guarantees were demonstrated for every scenario.
SP4	Resource utilization	Yes	Green AI optimization mechanisms, benchmarking services, resource monitoring, workload scheduling, and resource allocation capabilities were implemented through RLFEED, ORCH, and RSCMANAGER.
SP5	Interoperability	Partially	REST APIs and modular architecture provide interoperability, but complete interoperability with every external or legacy system cannot be guaranteed.
SP6	Operability	Yes	The Visualization Engine provides a unified interface for workflow configuration, execution, monitoring, explainability, and reporting, enabling practical operation of the platform.
SP7	User engagement	Yes	Human-in-the-Loop mechanisms, explainability services, visualization dashboards, feedback collection, and monitoring facilities actively support user participation throughout the AI lifecycle.
SP8	Availability	Yes	Kubernetes-based deployment, containerized services, distributed storage mechanisms, and orchestration services support reliable platform operation and service availability.
SP9	Confidentiality	Partially	Authentication, RBAC, federated learning and secure storage exist, but no complete end-to-end security certification or penetration testing demonstrating confidentiality across every deployment environment.
SP10	Integrity	Yes	Data management services, monitoring mechanisms, blockchain-based traceability, and controlled workflow execution support data and model integrity throughout the platform lifecycle.
SP11	Authenticity	partially	Authentication exists (Keycloak, RBAC), but authenticity in the broader sense (trusted identities across heterogeneous external infrastructures) depends on external identity providers and deployment policies.
SP12	Modularity	Yes	The final architecture follows a modular service-oriented design composed of independently deployable yet interoperable platform components coordinated through ORCH.

NFR ID	Title	Satisifed?	Validation evidence
SP13	Testability	Yes	Component-level implementation, platform integration activities, pilot deployment, validation procedures, and KPI-based evaluation activities provide evidence of successful testing and validation.
SP14	Adaptability	Yes	The platform supports heterogeneous data modalities, multiple deployment environments, configurable workflows, adaptive optimization services, and pilot-specific AI use cases.

6 Conclusions

This deliverable presented the final architecture of the RAIDO platform and documented its evolution from the initial architectural blueprint introduced in D2.3 to a fully integrated, deployed, and validated operational ecosystem. Through the implementation, integration, deployment, and validation activities performed throughout the project, the proposed architectural design has been successfully realized into a platform capable of supporting end-to-end AI optimization workflows across heterogeneous application domains and computing environments.

The final platform consolidates the technological outcomes developed across the project into a cohesive ecosystem that supports data management, AI optimization, workflow orchestration, trustworthiness assessment, deployment, resource management, monitoring, and governance within a unified operational environment. The successful realization of this architecture demonstrates the maturity of the developed platform and confirms its capability to support reliable, trustworthy, explainable, and sustainable Artificial Intelligence applications. The following sections summarize the principal achievements of the final platform, highlight its architectural evolution since Deliverable D2.3, discuss the validation of the proposed architecture against the project requirements, and outline future directions for the continued evolution of the RAIDO ecosystem.

6.1 Final platform achievements

One of the principal achievements of the RAIDO project has been the successful realization of a modular, interoperable, and operational AI platform that integrates the technological outcomes developed across the project into a unified ecosystem. Rather than operating as isolated services, the individual platform components have been successfully integrated through a common architectural framework, enabling the coordinated execution of data-centric and model-centric AI optimization workflows.

The resulting platform combines advanced data management, AI optimization, workflow orchestration, trustworthiness assessment, deployment management, resource allocation, monitoring, and governance capabilities within a common execution environment. This integrated approach enables the efficient management of datasets, AI models, metadata, monitoring information, benchmarking results, and optimization artefacts while supporting the execution of AI workflows across cloud, edge, and hybrid computing environments.

The flexibility of the final architecture has been demonstrated through its successful application across the diverse RAIDO pilot domains, including smart energy systems, smart farming, healthcare and pharmacogenomics, and industrial robotics. The coordinated operation of the platform services across these heterogeneous application

scenarios confirms both the interoperability of the developed ecosystem and its capability to address domain-specific requirements without requiring fundamental architectural modifications.

The successful integration, deployment, and validation of the platform components demonstrate the operational maturity of the final RAIDO architecture and confirm its capability to support scalable, trustworthy, explainable, and sustainable AI workflows. Consequently, the platform establishes a robust technological foundation for future research activities, industrial exploitation, and the continued evolution of reliable and sustainable Artificial Intelligence solutions.

6.2 Architectural evolution since D2.3

Deliverable D2.3 established the initial architectural vision of the RAIDO platform, defining the overall system structure, the principal technological components, and the interactions required to support the project objectives. At that stage, the architecture represented a conceptual framework intended to guide the subsequent implementation and integration activities performed throughout the project.

Since the publication of D2.3, substantial progress has been achieved across all architectural domains. The conceptual components defined during the design phase have been fully implemented, integrated, deployed, and validated within a common operational platform. In parallel, the architectural framework has been enriched with the backend services, standardized communication interfaces, orchestration mechanisms, deployment infrastructure, monitoring capabilities, security services, and governance mechanisms required to support the practical operation of the platform.

The architecture has also evolved to address the technical challenges and operational requirements identified during the implementation, integration, and pilot execution phases. Continuous refinements introduced throughout the project improved the interoperability, scalability, usability, resource management, and deployment capabilities of the platform while preserving the modular principles established during the initial architectural design.

Consequently, the final architecture presented in this deliverable represents not only the successful realization of the design introduced in Deliverable D2.3 but also its evolution into a mature, operational, and validated ecosystem capable of supporting end-to-end AI optimization workflows across heterogeneous computing environments.

6.3 Architectural validation and requirements fulfilment

The successful realization of the RAIDO platform demonstrates that the architectural objectives established during the early phases of the project have been effectively achieved. Throughout the implementation, integration, deployment, and pilot validation activities, the proposed architecture was progressively verified to ensure that the

developed platform remained aligned with the stakeholder requirements, functional requirements, and non-functional requirements defined in Deliverables D2.1 and D2.3.

As demonstrated in Chapter 5, the updated Requirements Traceability Matrix establishes direct traceability between the initial project requirements and the implemented platform services, providing evidence that the architectural capabilities envisioned during the design phase have been successfully realized. The traceability analysis, together with the deployment and validation activities documented in Deliverable D6.1, confirms the operational implementation of the proposed architecture and demonstrates that the final platform satisfies the vast majority of the identified functional and non-functional requirements.

The validation activities further confirmed the interoperability of the platform services, the effectiveness of the deployment and orchestration mechanisms, the scalability of the distributed execution environment, and the successful operation of the integrated AI optimization workflows across the RAIDO pilot domains. The assessment also demonstrated that the architectural principles adopted throughout the project—including modularity, standardized interfaces, centralized data management, workflow orchestration, and intelligent resource management—provide a robust foundation for the reliable operation of the platform.

Overall, the validation evidence presented throughout this deliverable confirms that the final RAIDO platform successfully fulfills the architectural vision established at the beginning of the project. The implemented ecosystem therefore provides a mature, interoperable, and operational framework capable of supporting reliable, trustworthy, explainable, and sustainable Artificial Intelligence applications while establishing a solid foundation for future technological evolution and industrial adoption.

6.4 Future outlook

Although the objectives of the RAIDO project have been successfully achieved, the final platform has been intentionally designed to support future technological evolution beyond the project lifetime. Its modular service-oriented architecture, standardized communication interfaces, and distributed deployment model facilitate the integration of new AI optimization techniques, explainability methods, trustworthiness services, and domain-specific capabilities without requiring fundamental architectural modifications. These characteristics provide the flexibility required to accommodate future advances in Artificial Intelligence and evolving application requirements.

Beyond the application domains validated during the project, the RAIDO platform provides a reusable technological foundation that can be adapted to additional sectors requiring reliable, trustworthy, and sustainable AI solutions. The combination of intelligent resource management, scalable deployment mechanisms, and interoperable platform services enables the architecture to support future research activities,

industrial exploitation, and the continuous adoption of AI technologies across heterogeneous computing environments.

The resulting architecture provides a robust technological foundation for the future evolution of trustworthy and sustainable AI systems. Its modular design, standardized communication mechanisms, and service-oriented organization enable the incorporation of new AI technologies, optimization methods, and deployment environments with minimal architectural modifications, facilitating both future research activities and industrial adoption across diverse application domains.

Funded by
the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101135800