H2020 - INDUSTRIAL LEADERSHIP - Information and Communication Technologies (ICT)

ICT-38-2020: Artificial intelligence for manufacturing

Grant Agreement: 957362

Explainable Manufacturing Artificial Intelligence

WP5: XMANAI Platform Continuous Integration

D5.1: System Architecture, Bundles Placement Plan and APIs Design

Deliverable Leader: Suite5

Due Date: 30/10/2021

Dissemination Level: Public

Version: F1.00

Short Abstract

This deliverable provides the design of the XMANAI reference architecture by: (a) designating the architecture blueprints across the tiers, services bundles, components and application perspectives, (b) designating the core workflows along the user journeys in Explainable AI for business users, data scientists and data engineers, for the interaction among the different XMANAI components, (b) defining the core functionalities, the mapping to the XMANAI technical requirements and MVP features, and the main interactions of the XMANAI components that shall be delivered in the XMANAI Centralized Cloud and On-Premise (Private Cloud) installations, (c) obtaining some early perspectives on the features of the XMANAI manufacturing apps and their alignment with the business requirements of the XMANAI demonstrators.

Further Information: www.ai4manufacturing.eu

Document Log

Contributors	**Suite5**, TXT, UBITECH, Fraunhofer, TYRIS, INNOVALIA, ATHENA, KBIZ
Internal Reviewer 1	Fraunhofer
Internal Reviewer 2	UBITECH
Type	Report

History

Versions	***Description***
D0.1	Initial Table of Contents
D0.2	Initial draft of Section 2 consolidating inputs
D0.3	Initial draft of Section 3
D0.4	Initial inputs to Sections 4 and 5
D0.5	Updated version of Sections 3-5
D0.6	Consolidated Section 1 and 6
D0.7	Full Deliverable Draft sent for internal review
R0.1	Revision of internal reviewer 1 (Fraunhofer)
R0.2	Revision of internal reviewer 2 (UBITECH)
D0.8	Updated version addressing comments received during the internal review
F1.0	Final version submitted to the EC

Executive Summary

The XMANAI Deliverable D5.1 “System Architecture, Bundles Placement Plan and APIs Design” documents the results acquired during the implementation of task T5.1 “Platform Architecture, Bundles Communication Design and APIs Definition” till M12. The purpose of this deliverable is to define the reference architecture of the whole XMANAI platform in the form of architecture blueprints of the components and manufacturing apps that will drive the next implementation steps of the project. The XMANAI reference architecture is elaborated based on four (4) perspectives:

- The tier perspective that anticipates the co-existence of the XMANAI (Centralized) Cloud Infrastructure, the XMANAI On-Premise (Private Cloud) Environments and the XMANAI Manufacturing Apps Portfolio.
- The services bundle perspective, including eight (8) services bundles, namely the Data Collection & Governance Services, the Scalable Storage Services, the Data Manipulation Services, the XAI Model Lifecycle Management Services, the XAI Execution Services, the XAI Insights Services, the Secure Asset Sharing Services and the Management Services Bundle.
- The component perspective that extends over twelve (12) main components at high level, including: the Data Access Manager, the Data Handler, the Data Manipulation Engine, the Execution & Orchestration Engine, the Identity & Authorisation Manager, the Knowledge Graph Manager, the Model Engineering Engine, the Provenance Engine, the XAI Marketplace, the XAI Models Catalogue, the XAI Pipeline Manager, and the XAI Visualization Engine.
- The applications perspective that refers to the manufacturing apps addressed to solving concrete manufacturing problems and use cases for the XMANAI demonstrators, including the XMANAI-PrOp app for Production Optimization, the XMANAI-PDeF app for Product Demand Forecasting, the XMANAI-PPQO app for Process/Product Quality Optimization and the XMANAI-SAMP app for Smart Semi-Autonomous Hybrid Measurement Planning.

Taking into consideration the XMANAI high-level usage scenarios, noted as User Journeys, that were designated for the Business User, the Data Scientist and the Data Engineer in the XMANAI Deliverable D1.2, a set of four (4) workflows (referring to the collaborative AI preparation, AI experimentation, AI insights extraction and AI application workflows) has been prescribed in the form of BPMN diagrams to highlight the anticipated interactions among the XMANAI components and services.

In addition, the XMANAI Platform Components have been specified through an iterative process among all technical partners to ensure that the needs of the intended, target users (business users, data scientists and data engineers in manufacturing) are appropriately addressed. For each component, its functionalities have been defined, its sub-components (that are analyzed in detail in the XMANAI Deliverables D2.1 and D3.1) have been identified, its mapping to the XMANAI technical requirements and MVP features is performed and its interactions have been designed through UML sequence diagrams. The functionalities of the XMANAI manufacturing apps have been also elaborated while particular attention has been paid on their alignment to the business requirements expressed by the corresponding demonstrators.

The results of this deliverable have been prepared hand-in-hand, in a consistent way, with the advancements in the detailed specifications of the Data and AI Services Bundles in WP2 “Industrial Asset Management and Secure Asset Sharing Bundles” and WP3 “Core Artificial Intelligence Bundles for Algorithm Lifecycle Management”. The XMANAI reference architecture shall guide the early development tasks across all work packages (WP2-WP6). Finally, the T5.1 activities related to this deliverable will continue to reflect on the project’s advancements and shall proceed to appropriate revisions and updates on the reference architecture as required, that shall be documented in conjunction with the different XMANAI Platform releases (i.e. alpha release on M21 – D5.2, beta release on M32 – D5.3, release 1.0 on M42 – D5.4).

Table of Contents

List of Figures

List of Tables

1 Introduction

This section provides a brief introduction to this deliverable.

1.1 XMANAI Project Overview

Despite the indisputable benefits that Artificial Intelligence (AI) can bring in society and in any industrial activity, humans typically have little insight about AI itself and even less concerning the knowledge on how AI systems make any decisions or predictions due to the so-called “black-box effect”. Many of the machine learning/deep learning algorithms are opaque and not possible to be examined after their execution to understand how and why a decision has been made. In this context, to increase trust in AI systems, XMANAI aims at rendering humans (especially business experts from the manufacturing domain) capable of fully understanding how decisions have been reached and what has influenced them.

Building on the latest AI advancements and technological breakthroughs, XMANAI shall focus its research activities on Explainable AI (XAI) in order to make the AI models, step-by-step understandable and actionable at multiple layers (data-model-results). The project will deliver "glass box" AI models that are explainable to a "human-in-the-loop", without greatly sacrificing AI performance. With appropriate methods and techniques to overcome data scientists’ pains such as lifecycle management, security and trusted sharing of complex AI assets (including data and AI models), XMANAI provides the tools to navigate the AI’s “transparency paradox” and therefore:

(a) accelerates business adoption addressing the problematic that “if manufacturers do not understand why/how a decision/prediction is reached, they will not adopt or enforce it”, and

(b) fosters improved human/machine intelligence collaboration in manufacturing decision making, while ensuring regulatory compliance.

XMANAI aims to design, develop and deploy a **novel Explainable AI Platform** powered by explainable AI models that inspire trust, augment human cognition and solve concrete manufacturing problems with value-based explanations. Adopting the mentality that “AI systems should think like humans, act like humans, think rationally, and act rationally”, a catalogue of **hybrid and graph AI models** is built, fine-tuned and validated in XMANAI at 2 levels: (i) baseline AI models that will be reusable to address any manufacturing problem, and (ii) trained AI models that have been fine-tuned for the different problems that the XMANAI demonstrators’ target. A bundle of **innovative manufacturing applications and services** are also built on top of the XMANAI Explainable AI Platform, leveraging the XMANAI catalogue of baseline and trained AI models.

XMANAI will validate its AI platform, its catalogue of hybrid and graph AI models and its manufacturing apps in **4 realistic, exemplary manufacturing demonstrators** with high impact in: (a) optimizing performance and manufacturing products’ and processes’ quality, (b) accurately forecasting product demand, (c) production optimization and predictive maintenance, and (d) enabling agile planning processes. Through a scalable approach towards Explainable and Trustful AI as dictated and supported in XMANAI, manufacturers will be able to develop a robust AI capability that is less artificial and more intelligent at human and corporate levels in a win-win manner.

1.2 Deliverable Purpose and Scope

This deliverable aims at designing and documenting the XMANAI reference architecture that will provide the basis for the detailed specification, development and integration activities of the overall XMANAI platform along its Data and AI related Services Bundles, its XAI Algoritmms and Models Catalogue, and the Manufacturing Apps. Building on the XMANAI technical requirements and the MVP features, the XMANAI architecture design involves the following activities:

- **Study of different manufacturing initiatives** such as RAMI 4.0 and IIRA, to ensure that XMANAI takes into account the structure and guidelines of the specific reference models and frameworks for digital manufacturing platforms.
- **High-level design of the XMANAI architecture across** different perspectives, namely: (a) the **tiers perspective** in order to anticipate where and how the XMANAI Platform is delivered and deployed, (b) the **services bundle perpective** that abstracts how the data and XAI-related services are developed, (c) the **component perspective** that outlines the main functionality of the different XMANAI components and how they are brought together in the different tiers and services bundles, and (d) the **application perspective** that highlights the scope of the manufacturing apps developed per demonstrator, along with their expected interactions with the XMANAI Platform.
- **Workflow design of the combined phases** that can be extracted **from the business user journey, the data scientist journey and the data engineer journey**, and refer to the planned integration of different XMANAI components to deliver the expected functionality over the AI preparation, the AI experimentation, the AI insights and the AI application phases.
- **Specification of the XMANAI platform components** by outlining their main functionalities, identifying the sub-components they include (that will be detailed in the XMANAI Deliverables D2.1 and D3.1), ensuring their alignment with the XMANAI technical requirements and MVP features, and defining their expected interactions with other components.
- **Specification of the XMANAI manufacturing apps** by outlining their main functionalities, revisiting their alignment with the XMANAI business requirements, and defining their expected interactions with the XMANAI Platform.

Since the architecture design and specification of the XMANAI platform is an ongoing activity following the agile software development pattern through the XMANAI project implementation, any updates and revisions that are considered as relevant to be introduced in the XMANAI reference architecture will be reported with the different releases of the XMANAI Platform (i.e. alpha release on M21 – D5.2, beta release on M32 – D5.3, release 1.0 on M42 – D5.4).

1.3 Impact and Target Audiences

The XMANAI reference architecture is instrumental for driving the XMANAI platform and apps delivery, providing their initial positioning, scope and interactions. In general, it is addressed to more technical audiences within the XMANAI consortium, namely the scientific and technical partners which are involved in the research, development, integration and deployment activities of the XMANAI Platform and/or manufacturing apps. However, it needs to be noted that the XMANAI reference architecture is also relevant for the business users in order to obtain a broader understanding of the expected XMANAI advancements.

1.4 Deliverable Methodology

In XMANAI, a solid methodological approach for the architecture design has been meticulously followed to ensure that the deriving reference architecture effectively builds on the XMANAI background, addresses concrete business and technical needs of its relevant stakeholders and is formalized according to a set of basic principles. In particular, the XMANAI architecture that is documented in this deliverable builds on the premises created by the XMANAI MVP and technical requirements described in the XMANAI Deliverable D1.2, while considering the preliminary architecture that was provided in the XMANAI DoA. The steps that were followed included collaborative work among all involved partners (in WP5, but also in WP2, WP3, WP4 and WP6):

I. Elaboration on the tiers and services bundles perspective of the XMANAI platform to effectively anticipate different options for manufacturers, including the XMANAI demonstrators, that may have different policies and needs when it comes to data security and infrastructure availability. When defining the different perspectives in XMANAI, the different RAMI 4.0 layers (i.e. business, functional, information, communication, integration, assets

layers) and IIRA viewpoints (i.e. Business Viewpoint, Usage Viewpoint, Functional Viewpoint, Implementation Viewpoint) have been considered to ensure that all necessary design aspects have been appropriately reflected (collectively in the XMANAI Deliverables D1.2, D2.1, D3.1 and D5.1).

II. Definition of a preliminary list and positioning of the XMANAI components, including iterative discussions on their scope and main functionalities. It needs to be noted that many iterations occurred while trying to rationalize the large number of components that were included in the preliminary architecture as defined in the DoA, while ensuring that the functionalities that have emerged from the elicitation of the XMANAI technical requirements and MVP have been anticipated in the different components.

III. Design of the workflows that emerge from the data scientist, business user and data engineer journeys in order to define the interrrelations between the XMANAI Platform components at high level.

IV. Iterative specification of the XMANAI Platform components, designating their architecture blueprints and identifying complementarities and missing functionalities in respect to the prioritized MVP features and their associated technical requirements.

V. Design of the XMANAI manufacturing apps on the basis of the business requirements expressed by the XMANAI Demonstrators.

Since XMANAI will follow an agile development mentality, the driving principles behind its reference architecture include: (a) clear separation of concerns, (b) design for flexibility and change, (c) reliance on open source and partners' technologies.

1.5 Dependencies in XMANAI and Supporting Documents

The XMANAI Deliverable D5.1 reports the results of Task T5.1 "Platform Architecture, Bundles Communication Design and APIs Definition" in the context of the WP5 "XMANAI Platform Continuous Integration" activities. D5.1 has been prepared in close collaboration with the design activities performed in "WP2 – Industrial Asset Management and Secure Asset Sharing Bundles" and "WP3 - Core Artificial Intelligence Bundles for Algorithm Lifecycle Management" that are documented in the XMANAI Deliverables D2.1 and D3.1.

In essence, D5.1 has built on the outcomes of Tasks T1.3 "Platform Requirements Elicitation, Data Acquisition and AI Scenarios" and T1.4 "XMANAI Concept Elaboration, MVP Definition and Validation" reported in D1.2. In addition, the early design of the manufacturing apps has been prepared in WP6 "Demonstrators Setup, Operation and Business Value Exploration" and documented in this deliverable, taking into consideration the business requirements collected in the framework of T6.1 "Demonstrators Requirements Elicitation" (as described in D6.1).

Finally, this deliverable is expected to provide the architecture blueprints behind the implementation activities to be performed across "WP2 - Industrial Asset Management and Secure Asset Sharing Bundles", "WP3 - Core Artificial Intelligence Bundles for Algorithm Lifecycle Management", "WP4 - Novel Artificial Intelligence Algorithms for Industrial Data Insights Generation", "WP5 - XMANAI Platform Continuous Integration" and "WP6 - Demonstrators Setup, Operation and Business Value Exploration".

1.6 Document Structure

The structure of this document is organized as follows:

- Section 2 outlines the XMANAI reference architecture, including its design from different perspectives and abstraction levels.
- Section 3 provides the basic workflows that can be extracted from the different user journeys for the core XMANAI stakeholders (business users, data scientists and data engineers) and involve interaction between different components in the XMANAI architecture.

- Section 4 documents the functionalities, the associated technical requirements and the interactions of the main XMANAI platform components.
- Section 5 reports on the features and the relevant business requirements of the XMANAI manufacturing apps while discussing their expected interactions with the XMANAI platform to ensure generalization and applicability of the overall approach.
- Section 6 summarises the key aspects regarding the architecture design activities and provides an outlook to the next steps.
- Annex I lists the references.

2 XMANAI Reference Architecture

This section presents the XMANAI reference architecture from four (4) perspectives: (a) the tiers perspective that clarifies where the overall XMANAI platform is conceptually placed, (b) the services bundles perspective that provides an abstraction layer for the different services and components that constitute the XMANAI architecture, (c) the components perspective that presents the list of software artifacts that will be delivered in XMANAI throughout the activities of the core technical WPs (WP2-WP5), and (d) the applications perspective that highlights the different manufacturing apps that will be delivered per demonstrator in WP6.

2.1 Tiers Perspective

Taking into consideration the underlying business requirements for data security and isolation per demonstrator and the diverse manufacturing problems that are to be addressed in XMANAI in a generic, reusable manner, the XMANAI reference architecture has been conceptually divided in three tiers (as depicted in the following figure):

- The XMANAI Cloud Infrastructure lies at the core of the whole XMANAI MVP and essentially represents the centralized cloud instance of the XMANAI Platform, including: (a) the **Core AI Management Platform** that brings together all the XMANAI functionalities and is responsible for the design of the data and / or AI pipelines and the orchestration of their execution; and (b) the **Secure Execution Clusters (SEC)** that are triggered/spawn on demand by the Core AI Management Platform, for executing data and / or AI pipelines on an isolated, per organization basis.
- The XMANAI On-Premise Environments that essentially represent the parts of the XMANAI Platform that can be hosted and executed in a private cloud instance of a demonstrator / manufacturer. Such a deployment is considered as necessary for manufacturers who have imposed restrictions that their data are not leaving their premises or their own cloud infrastructures, in general.
- The XMANAI Manufacturing Apps Portfolio that essentially offers AI manufacturing intelligence solutions that are targeted to the needs of the different manufacturers through customized dashboards, mobile apps or AR apps depending on the manufacturing problem at hand. Each application needs to seamlessly communicate with the XMANAI Cloud Infrastructure and/or the XMANAI On-Premise Environments in order to retrieve the appropriate data or AI pipeline execution results.

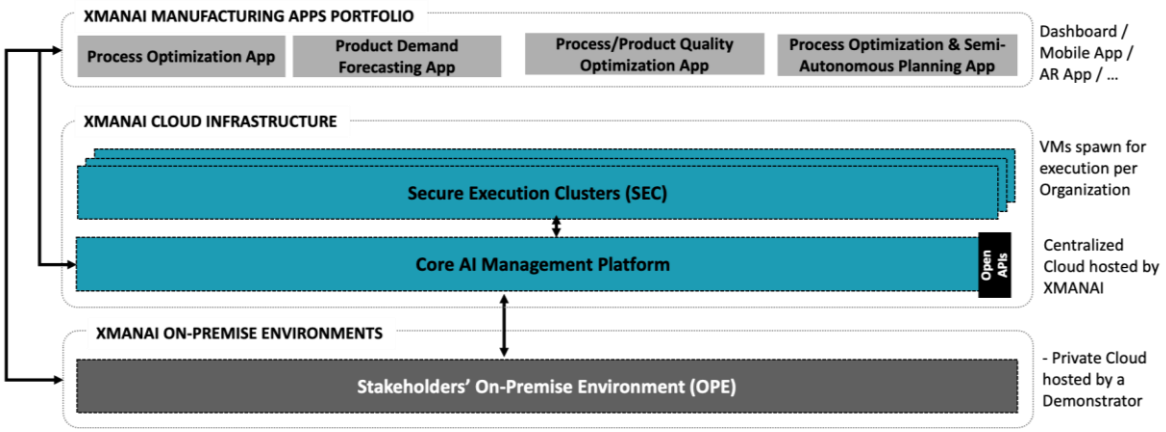

Figure 2-1: XMANAI Reference Architecture – Tiers Perspective

2.2 Services Bundles Perspective

The XMANAI reference architecture consists of a number of software components and services whose functionalities can be abstracted in the following services bundles that can be placed in the different tiers:

- **Data Collection & Governance Services** *(available in the XMANAI Cloud Infrastructure and the XMANAI On-Premise Environments)* that practically refer to how data can be consistently collected and managed, through the configuration and execution of appropriate data handling processes. The Data Collection Services Bundle is practically responsible for the secure and reliable admission of a data asset in the centralized XMANAI cloud or the on-premise (private cloud) installations according to a schedule (through APIs exposed by the legacy and operational systems of the manufacturer) or at batch level (as files extracted by the systems of the manufacturer) and for applying the related processing rules that have been defined at "design" time. Any data that are ingested in XMANAI (through the Data Collection Services) are practically handed over to the Data Governance Services that ensure the provenance of the data across their lifecycle in XMANAI.
- **Scalable Storage Services** *(available in the XMANAI Cloud Infrastructure and the XMANAI On-Premise Environments)* that provide different persistence modalities depending on the type of the data asset to be stored (e.g. raw data, XAI models, XAI pipelines, XAI experiments, XAI results along with their explanations, contracts, access policies, provenance information, user data) and the storage location (i.e. centralized XMANAI cloud, on-premise/private cloud installation). The Scalable Storage Services Bundle is also responsible for indexing the metadata of the different data assets for query optimization and data discoverability performance.
- **Data Manipulation Services** *(available in the XMANAI Cloud Infrastructure and the XMANAI On-Premise Environments),* essentially bringing the data explainability and feature engineering mentality to the XMANAI platform. Such services allow the users to derive the underlying knowledge and ensure a common understanding of the data leveraging the XMANAI data models, while allowing them to manipulate the data as they consider appropriate to gain new insights and to render them ML/DL-ready in order to be used in training XAI models and in executing XAI pipelines.
- **XAI Model Lifecycle Management Services** *(available in the XMANAI Cloud Infrastructure)* that are responsible for designing, tracking and collaborating on XAI pipelines. The XAI Model Lifecycle Management Services extend on all aspects relevant to the (re)training, explaining, experiment tracking, evaluating (both from performance and security perspectives) and packaging of XAI models. Through such services, a user is practically empowered to build on the XMANAI catalogue of XAI models and techniques that are appropriate to solve specific manufacturing problems and are offered: (a) out-of-the box as baseline AI models, and (b) in their pre-trained forms per manufacturer in order for them to feed their data and easily find unknown patterns or elicit appropriate explanations.
- **XAI Execution Services** *(available in the XMANAI Cloud Infrastructure and the XMANAI On-Premise Environments),* that are triggered, according to user-defined schedules, to run: (a) XAI models/pipelines experiments during the experimentation phase, and (b) XAI pipelines in the production phase, addressing the relevant performance, speed, efficiency, and reliability requirements. Such services are responsible for monitoring and tracking the execution status in the Secure Execution Clusters and/or the On-Premise Environments, and for ensuring the XAI model/pipeline results and associated metrics are appropriately stored in the Scalable Storage Services.
- **XAI Insights Services** *(available in the XMANAI Cloud Infrastructure and the XMANAI On-Premise Environments)* that provide the core human interaction interfaces with the business experts and data scientists for collaboration purposes and for gaining insights in different phases towards

extracting manufacturing intelligence. Such services offer intuitive dashboards and diagrams for understanding the data and the XAI model results along with the necessary explanations / insights, but also the experimentation process that is running under the hood.

- **Secure Asset Sharing Services** *(available in the XMANAI Cloud Infrastructure)***,** which handle the cataloguing and trusted sharing of data and AI models across different manufacturing organizations and/or users working on specific projects, taking into consideration their IPR. Although sharing within an organization is enabled by default through the Management Services Bundle, sharing data assets between different organizations builds on an effective asset contracting mechanism that puts in place smarts contracts, powered by blockchain technologies.
- **Management Services Bundle** *(available in the XMANAI Cloud Infrastructure)* which is responsible for controlling access over all data assets based on their providers' preferences and for governing and securing both each manufacturer's data and the users credentials/identities.

2.3 Components Perspective

The components and services that have been specified to deliver the intended data, model and results explainability in XMANAI cross-cut the different services bundles and the different tiers presented in the previous sections.

As depicted in Figure 2-2, the Data Collection & Governance Services include: (a) the **Data Handler** that provides the necessary functionalities for configuring and managing the manufacturing data ingestion and staging prior to the data storage, (b) the **Data Anonymizer** that is available only in the On-Premise Environments to help the manufacturer to anonymize any potentially data locally prior to making them available in XMANAI, (c) the **Provenance Engine** which is responsible for tracking the lineage and changes performed over data assets in time.

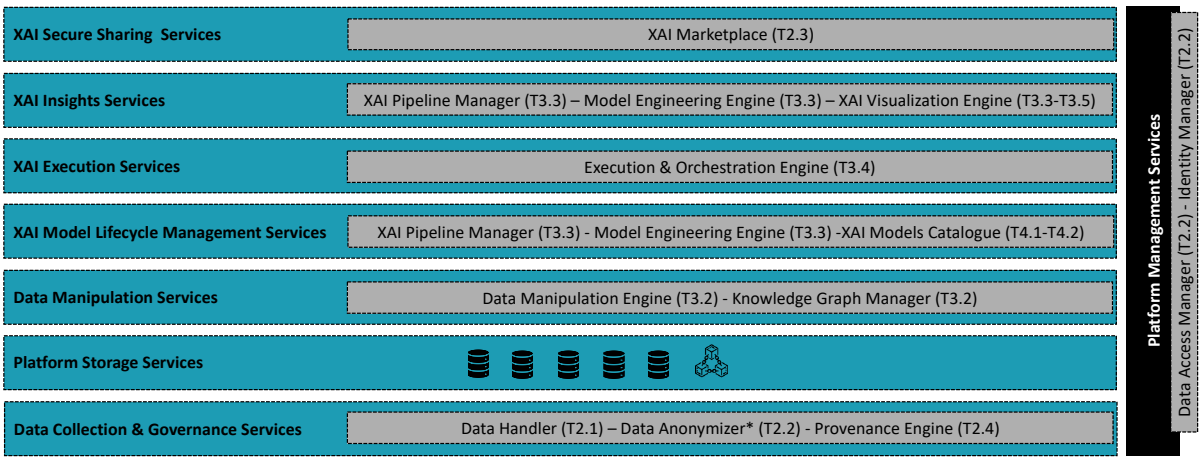

Figure 2-2: XMANAI Reference Architecture – Components Perspective

The Data Manipulation Services are delivered through: (a) the **Data Manipulation Engine** that allows data manipulation, augmentation and pre-processing, in general, in order to properly engineer and store the emerging features (to be used in XAI models/pipelines), and (b) the **Knowledge Graph Manager** that is responsible for handling and exposing the XMANAI data models, for validating the mapping of any dataset to the appropriate XMANAI data model and for retrieving the data in an appropriate representation from the Scalable Storage Services.

The XAI Model Lifecycle Management Services entail: (a) the step-by-step, collaborative design, validation and management of XAI pipelines bringing together different data preparation, model engineering and explainability functionalities through the **XAI Pipeline Manager**; (b) the training, management and evaluation activities related to XAI models as well as the experiment tracking and explanation generation functionalities through the **Model Engineering Engine**; and (c) the reuse and

applicability of different baseline algorithms (reused by popular AI libraries) and trained XAI models (for the manufacturing problems at hand in XMANAI) through the **XAI Models Catalogue**.

The XAI Execution Services are delivered through the **Execution & Orchestration Engine,** which is responsible for running any XAI pipeline designed in the XAI Model Lifecycle Management Services immediately or in scheduled dates.

The XAI Insights Services materialize mainly through the **XAI Visualization Engine** that provides appropriate visual insights on any data asset, including datasets, XAI models, XAI pipeline results and their associated explanations. The XAI Visualization Engine should not be considered as a stand-alone component, but as a component whose functionalities are delivered in collaboration with different components (e.g. the Model Engineering Engine, the XAI Pipeline Manager). Since the predictions need to be appropriately interpreted, certain preparatory work towards extracting XAI insights is also expected to be performed through the XAI Pipeline Manager and the Model Engineering Engine.

The expected functionalities of the XAI Secure Sharing Services are provided by the **XAI Marketplace**, which presents the data assets that are available in XMANAI by different manufacturers along with their metadata profile (e.g. license), once the applicable access policies for the organization/user that attempts to access them are resolved. Through the XAI Marketplace, the dynamic creation, negotiation, and finalization of non-monetary contracts for the acquisition of data assets is enabled. Such contracts are checked and enforced across all components of the XMANAI architecture.

Finally, the Platform Management Services include: (a) the **Data Access Manager** that enables the data asset providers to define and manage access policies to allow or deny authorization to a user/organization to access a specific data asset while enforcing such policies to all XMANAI components, and (b) the **Identity & Authorisation Manager** which provides the underlying organization and user management operations and is responsible for the authentication and authorization mechanisms of the XMANAI platform.

It needs to be noted that the Platform Storage Services do not include any components at the moment, but different data and metadata stores are created in collaboration between the different components of the rest of the data/XAI services bundles.

The XMANAI components are placed across the different tiers of the XMANAI reference architecture through the following ways:

- Centralized Cloud Only: There are components, such as the XAI Marketplace, the XAI Pipeline Manager, the XAI Models Catalogue, the Data Access Manager and the Identity & Authorisation Manager, that due to their scope and functionality must be always available only through the XMANAI Centralized Cloud Infrastructure.
- Combined Centralized Cloud and On-Premise: For most components, their provided functionalities are designed to be served both by the XMANAI Centralized Cloud Infrastructure and by the XMANAI On-Premise (Private Cloud) instances. In this case, typically the centralized cloud deployment offers the full functionality whereas the on-premise installation may offer partial or limited functionality for the specific components.
- On-Premise Only: The Data Anonymiser is in fact the only component in the XMANAI reference architecture that is available exclusively through the XMANAI On-Premise (Private Cloud) instances.

Figure 2-3 depicts the XMANAI components that shall become eventually available through the XMANAI Centralized Cloud Infrastructure. Their basic interactions are also depicted even though they will be detailed in Sections 3 and 4.

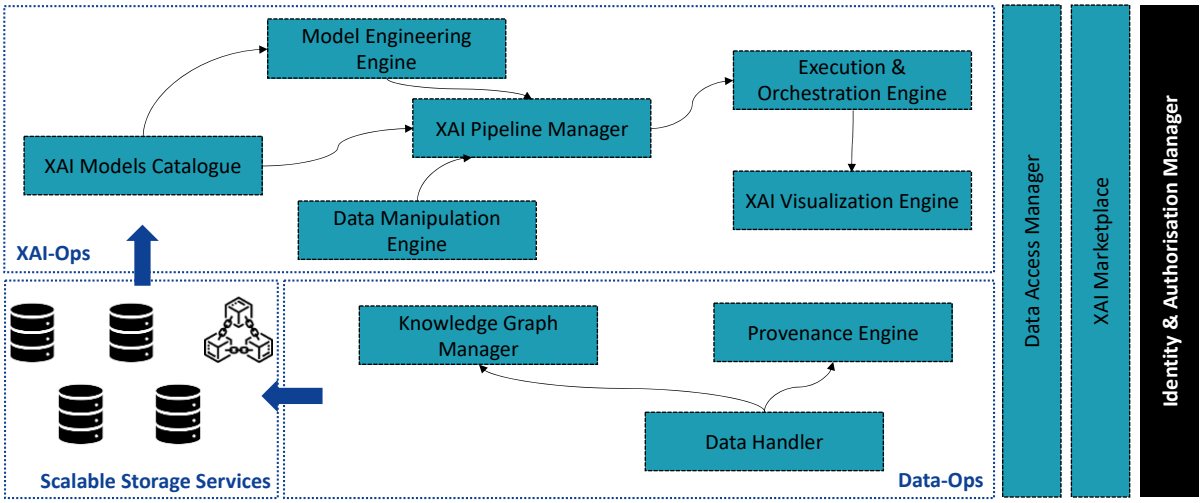

Figure 2-3: XMANAI Reference Architecture – Components Placement in the Centralized Cloud Infrastructure

Figure 2-4 depicts the XMANAI components that shall be part of the On-Premise (Private Cloud) installations for the manufacturers that are interested on this option. It needs to be clarified that the On-Premise Environments are not self-standing, but they need to interact with the Centralized Cloud Infrastructure in order to retrieve the configuration of the data and XAI pipelines that are to be executed.

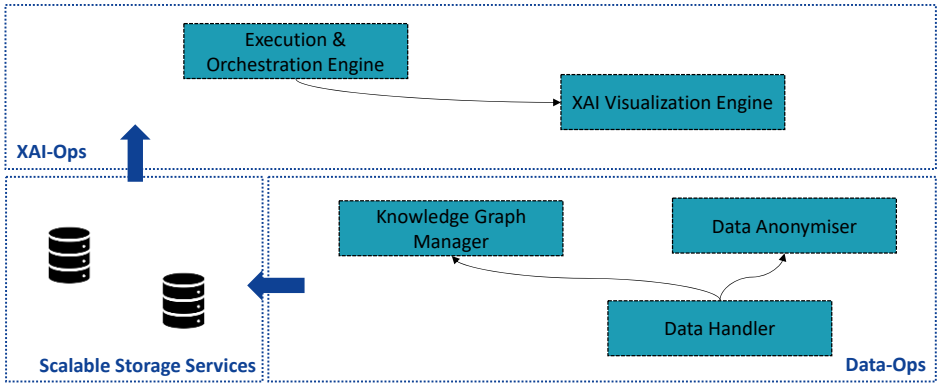

Figure 2-4: XMANAI Reference Architecture – Components Placement in the On-Premise Infrastructure

2.4 Applications Perspective

From an application perspective, a set of four (4) manufacturing applications will be developed to address specific manufacturing problems on which the XMANAI demonstrators focus:

- Production Optimization (related to Demonstrator I - FORD). In the XMANAI-PrOp app, an intuitive web interface will be indicatively provided to production managers, production coordinators, and workers (among other involved stakeholders) for automated intelligent planning of the factory production.
- Product Demand Forecasting (related to Demonstrator II - WHR). Through the XMANAI-PDeF app, different actors (e.g. central demand planners, D2C planners, D2C sales managers, etc) will have at their disposal dedicated demand forecasting interfaces to dive into the different predictions and explanations for the D2C channel.
- Process/Product Quality Optimization (related to Demonstrator III - CNH). The XMANAI-PPQO app will provide the production managers with web interfaces for “explainable” production scheduling, as well as the blue-collar workers with AR interfaces for collaborative maintenance.

- Smart Semi-Autonomous Hybrid Measurement Planning (related to Demonstrator IV - UNIMETRIK). The XMANAI-SAMP is essentially a layer over the M3 app that is already provided to the market by UNIMETRIK and shall provide improved guidance, recommendations and explanations over the best measurement plan parameters to the measurement managers and technicians.

In general, in order to display the appropriate data, XAI results/predictions and their associated explanations, all manufacturing apps need to be integrated with the XMANAI Platform in different ways:

I. To retrieve the necessary manufacturer's data that have been ingested in the XMANAI Platform and are available in the XMANAI Centralized Cloud or the On-Premise Environments.
II. To retrieve the results (in terms of predictions/decisions and their associated explanations) of XAI pipelines that have been configured to be executed in the XMANAI Centralized Cloud or the On-Premise Environments to solve the related manufacturing problems.
III. To leverage the single-sign on functionalities offered by the XMANAI Platform.

Along these directions, the XMANAI Platform essentially acts as: (I) a manufacturing data space that can be leveraged by any manufacturing app to acquire manufacturing data; (II) an XAI intelligence space that allows for collaborative experimentation and putting to production appropriate XAI pipelines to solve specific manufacturing problems; and (III) an identity provider that secures and safeguards the overall XMANAI operation.

3 User Journeys within the XMANAI Architecture

In Section 3, the different user journeys that were introduced in the XMANAI Deliverable D1.2 are revisited in order to showcase how they are supported by the different components of the XMANAI reference architecture.

3.1 Introduction

When it comes to bringing forward manufacturing intelligence through Explainable AI, the data scientists journey entails 5 phases, including: (DS.I) AI Preparation - Understand the data and the problem at hand; (DS.II) AI Experimentation - Prepare the data and handle problematic data cases; (DS.III) AI Experimentation - Collaboratively Design AI pipelines; (DS.IV) AI Experimentation - Explain AI models / results; and (DS.V) AI Experimentation - Evaluate AI models / pipelines, as explained in detail in the XMANAI Deliverable D1.2. The Business User Journey on its behalf includes 3 phases: (BU.I) AI Preparation - Provide and Understand Data; (BU.II) AI Experimentation - Contribute to better understanding and evaluating the AI models / pipelines / results; (BU.III) AI Insights - Understand AI results. Finally, the Data Scientist Journey on its behalf consists of 3 phases: (DE.I) AI Preparation - Collaborate on uploading / handling data; (DE.II) AI Experimentation - Collaborate on the design of AI pipelines; and (DE.III) AI Application - Deploy AI pipelines to production.

To this end, the different workflows that the XMANAI Platform will support have been designed to specify at high-level the expected interactions among the components, and include:

- XAI Preparation Workflow that involves business users, data scientists and data engineers and refers to DS.I, BU.I, and DE.I.
- XAI Experimentation Workflow that involves business users, data scientists and data engineers and refers to DS.II-DS.V, BU.II and DE.II.
- XAI Insights Workflow that involves business users, and data scientists and encapsulates the phases DS.IV, and BU.III.
- XAI Application Workflow that involves data scientists and, indirectly, the XMANAI Manufacturing apps and refers to DE.III.

Such workflows intend to provide indicative specifications about how the XMANAI Platform can be used to implement the different user journeys in order to guide the design and development activities, without though excluding additional interactions. It needs to be noted that the Scalable Storage Services and the Identity & Authorization Manager have been considered as implicit steps shared among all workflows and have not been intentionally depicted in their accompanying diagrams.

3.2 AI Preparation Workflow

In order to support data scientists, business users and data engineers to collaborate on the necessary steps to make available and comprehensive the manufacturing data for a business problem, different components of the XMANAI reference architecture need to interact as presented in the following figure. In particular, upon preparing the relevant data for the problem at hand “offline”, the business user and/or a data scientist access the XMANAI Centralized Cloud Infrastructure and create a new data harvesting process in the Data Handler. They proceed with the detailed configuration of the data harvesting process for the specific type of data (e.g. files or APIs) and configure the data mapping rules to make the data that will be uploaded more explainable with the help of the Knowledge Graph Manager (from which the details of the relevant XMANAI data model are retrieved). Then, in the Data Access Manager, they define the exact access policies that will be enforced every time a stakeholder may attempt to access the specific data. They also need to provide the metadata of the associated data asset for the specific data harvesting process in order to be used for cataloguing and easier search purposes in the XAI Marketplace. If they want, the business user and/or a data scientist are able to associate a data harvesting process with a data project in the Data Handler. When the execution time for the configured data harvesting process arrives (e.g. immediately for a file or according to a

schedule for API harvesting), the relevant service is triggered for execution in the cloud (in isolated spaces/VMs per organization) or in an on-premise environment (private cloud installation) depending on the specific manufacturer's (demonstrator's) preference. If the execution is successful, the data are stored (in the Scalable Storage Services) and tracked with the help of the Provenance Engine (e.g. for any changes performed if there is a recurring execution of the specific data harvesting process). Otherwise if the execution has failed, the error info is collected in order to help the business users and/or data engineers to understand what went wrong and correct it.

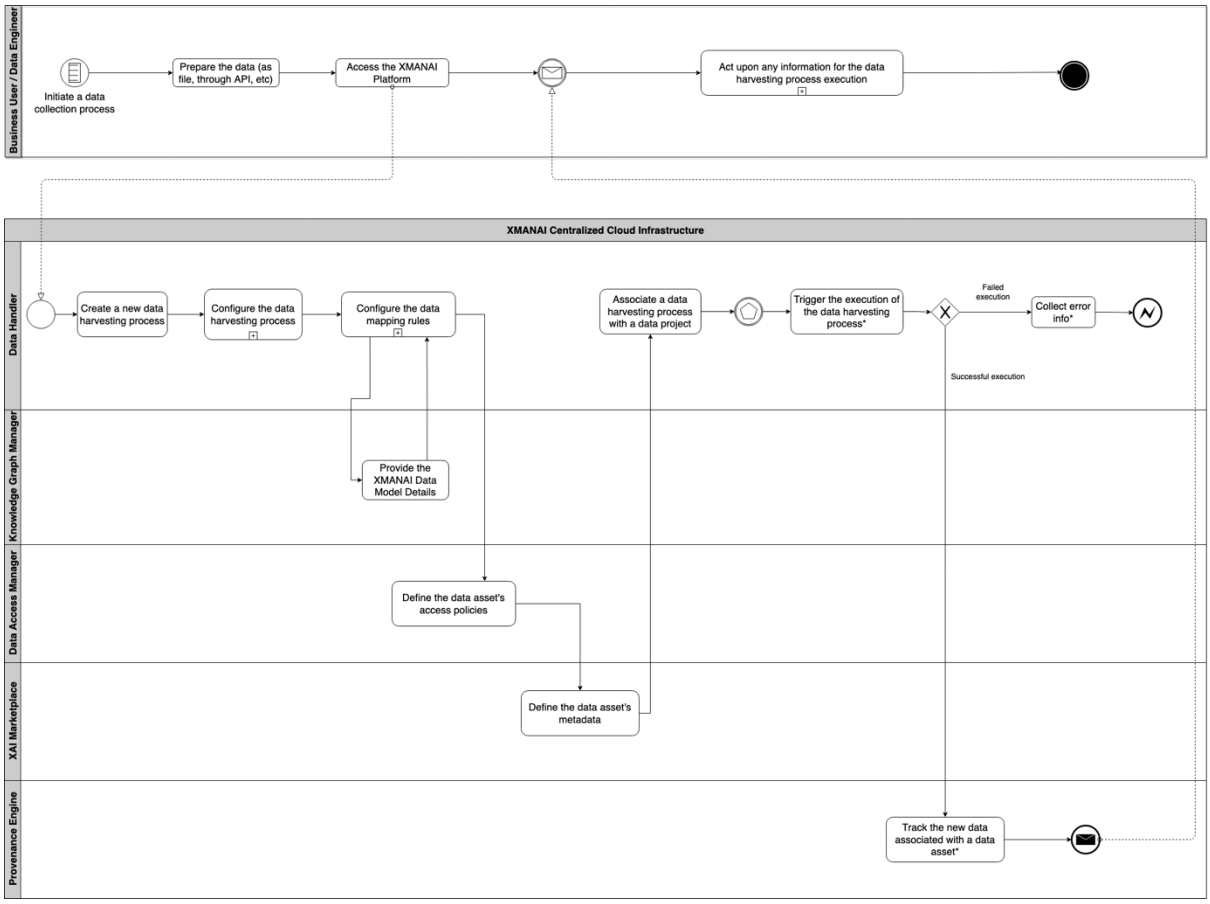

Figure 3-1: AI Preparation Workflow – Part I

Once the data are successfully harvested in the XMANAI platform, data scientists can access the XAI Marketplace in order to navigate to other data assets of interest for the problem at hand as depicted in Figure 3-2. They may search for specific data assets based on different metadata and view the results once the access policies are resolved in the Data Access Manager. The profile of the different data assets (e.g. datasets, models) is available through the XAI Marketplace yet if any additional information is required in the case of pre-trained models, it is acquired through the XAI Models Catalogue. If the data scientists find relevant data assets, they can acquire access to them through the sharing mechanism involving their providers and formalized through contracts. Once they have access to the data assets, if they are non-data assets (e.g. models), they may utilise them in the AI experimentation phase; otherwise, if they refer to datasets, they are able to view their structure and semantics with the help of the Knowledge Graph Manager. Then, they may perform data exploration in the Data Manipulation Engine and visualize the data with the help of the XAI Visualisation Engine, based on the data retrieved from the Scalable Storage Services with the help of the Knowledge Graph Manager. If the data scientists have any open questions for the data, they typically need to communicate with the business users, otherwise they may proceed to the AI experimentation phase.

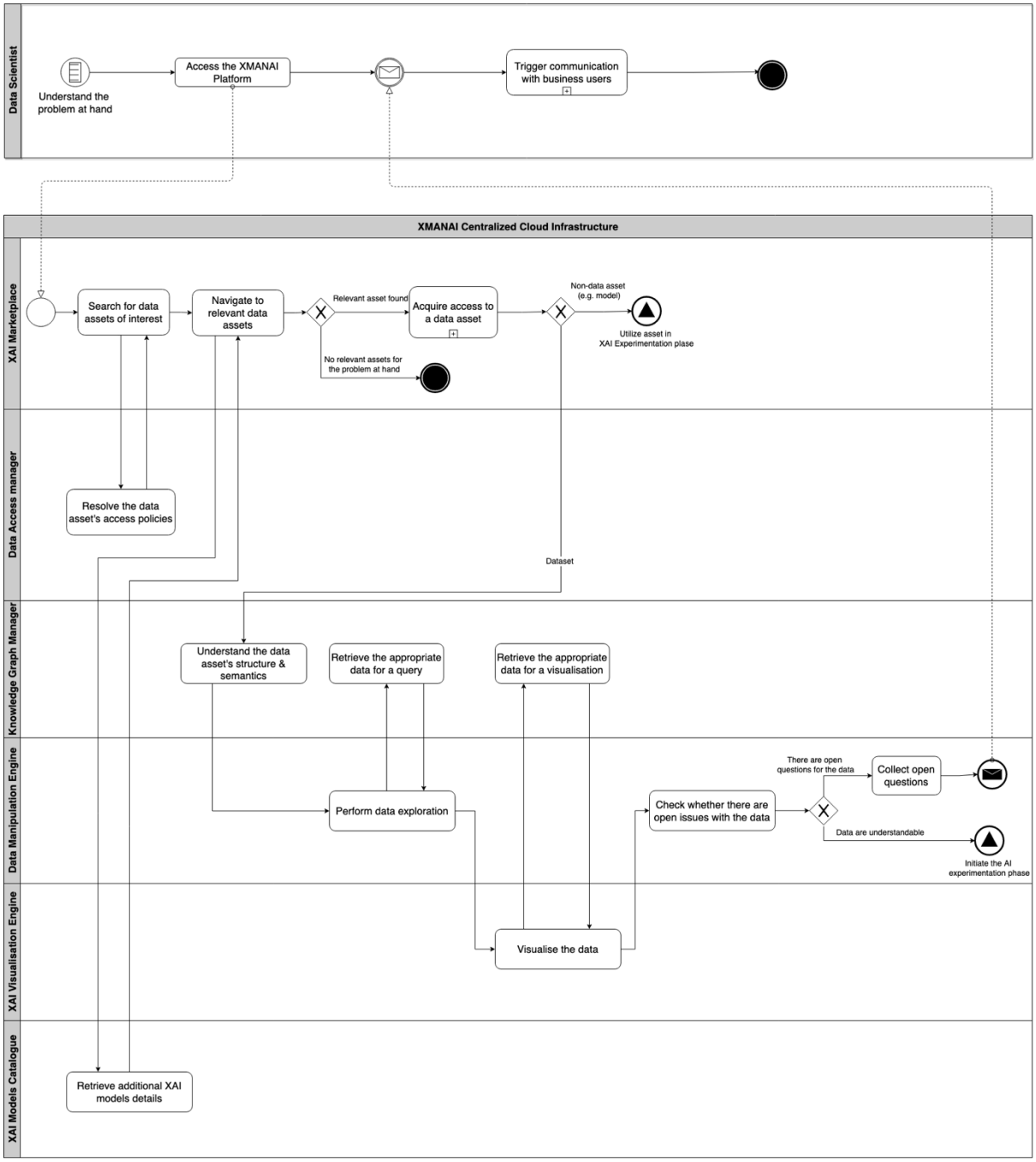

Figure 3-2: AI Preparation Workflow – Part II

3.3 AI Experimentation Workflow

During the AI Experimentation phase, the focus is mostly on the data scientists who need to consistently prepare the appropriate AI pipelines for the problem at hand. To this end, as depicted in Figure 3-3, they are able to design a new AI pipeline in the XAI Pipeline Manager and identify the relevant assets that shall be leveraged (once their access policies are resolved by the Data Access Manager). With the help of the Data Manipulation Engine, they are able to perform interactive exploration on the data to gain new insights while having at their disposal the dataset's structure and semantics (from the Knowledge Graph Manager). Once the data scientists are satisfied from the progress made, they can: (a) configure and view any visualisation chart of interest (with the help of the XAI Visualisation Engine), (b) proceed to the Model Engineering Engine for working on different AI models, (c) consolidate the data preparation functions that will be used in the pipeline configuration.

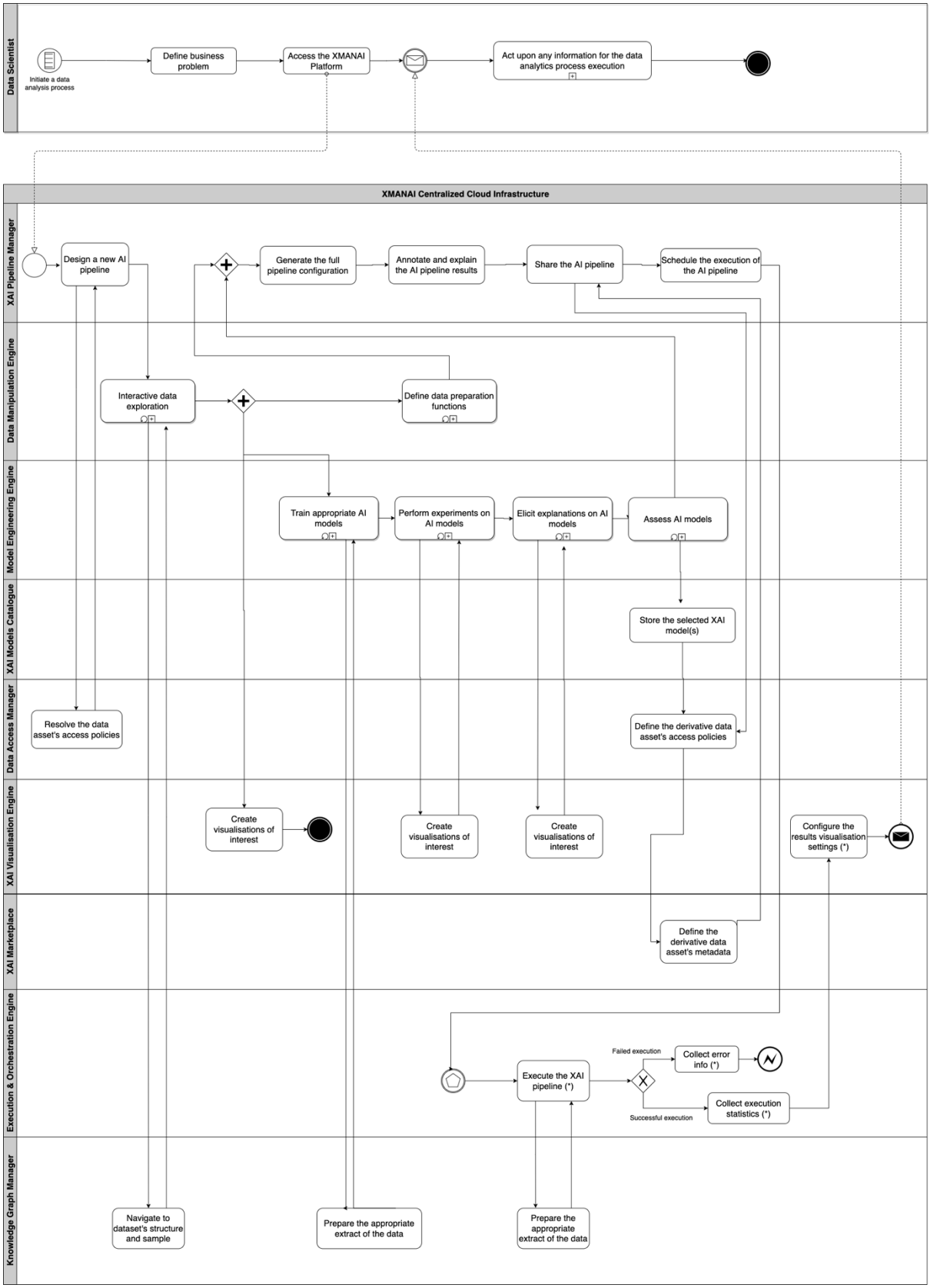

Figure 3-3: AI Experimentation Workflow

In case of proceeding with the AI models, the data scientists are able to select and train appropriate AI models for the problem at hand, perform experiments on the models to conclude on the optimal configuration (with the help of visualisations created through the XAI Visualisation Engine), elicit

explanations for the predictions of the selected model or for its operation as a whole in different ways and through different explainability techniques (again with the support of appropriate visualisations), and assess the performance and security of the XAI model. Once the trained XAI model is stored in the XAI Model Catalogue, the data scientist needs to define its access policies in the Data Access Manager and its metadata in the XAI Marketplace. Both the data preparation functions and the trained XAI models are included in the XAI pipeline configuration created by the XAI Pipeline Manager. The data scientists can attempt to annotate and explain the overall pipeline results before sharing the final pipeline in the XAI Marketplace (upon configuring the access policies and providing its metadata). It needs to be noted that at any moment during the design, the XAI pipeline can be also shared in draft form with business users and data engineers within the same organisation or working on the same project. Finally, the data scientists need to define the execution schedule of the XAI pipeline that becomes available (along with the pipeline configuration) to the Execution & Orchestration Engine that takes over its execution in the cloud or in an on-premise environment (private cloud installation) depending on where the specific manufacturer's (demonstrator's) data are stored.

3.4 AI Insights Workflow

When it comes to gaining appropriate AI insights in an understandable, explicit manner, the perspective of the business user is the most critical.

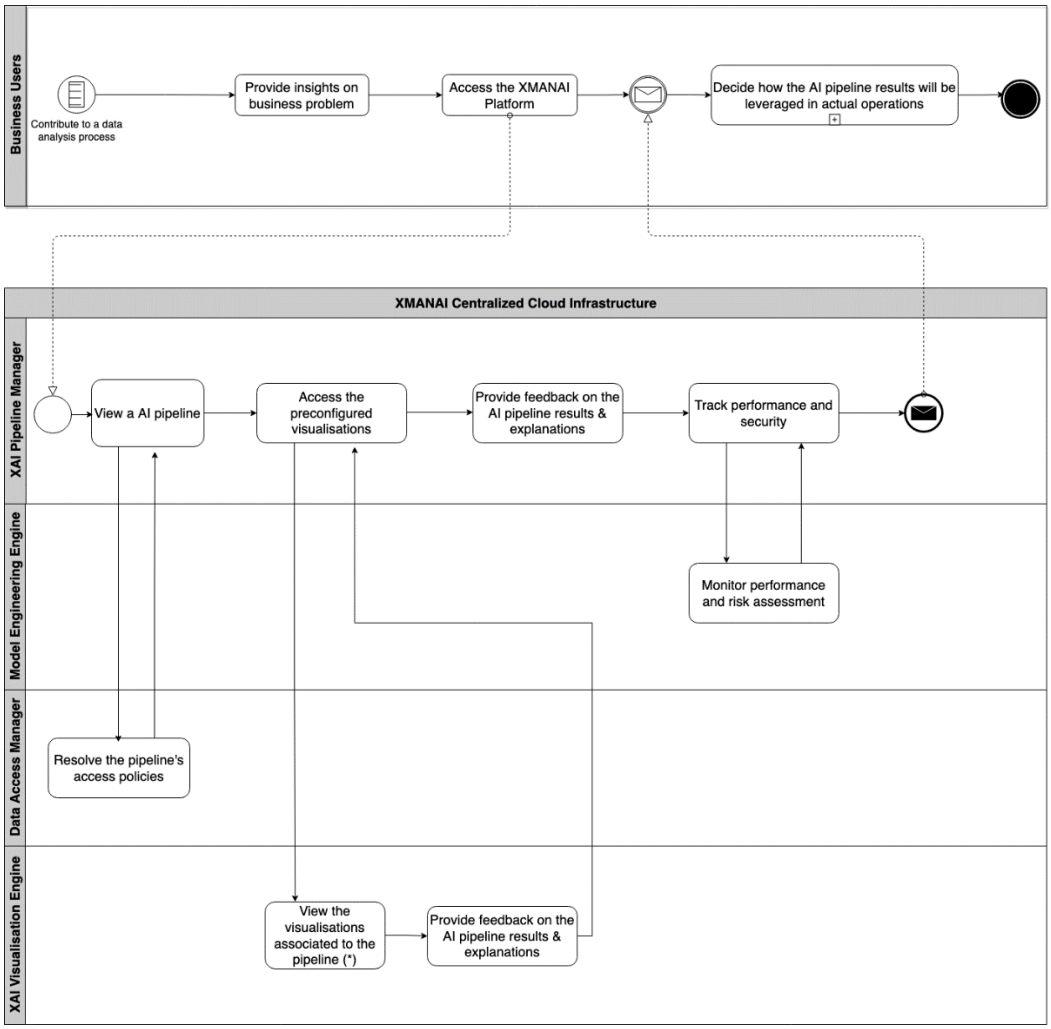

Figure 3-4: AI Insights Workflow

As depicted in Figure 3-4, the business users need to have a solid understanding of the business problem at hand so as to contribute whenever needed by accessing the XAI Pipeline Designer and viewing the relevant pipeline which they are eligible to access. They may view the preconfigured

visualisations that are associated to the specific XAI pipeline in the XAI Visualisation Engine (either in the centralized cloud or in the on-premise/private cloud installation) and provide their feedback. Further annotations and comments on the explanations associated with an XAI pipeline can be also generally provided in the XAI Pipeline Designer. The business users can also track the performance metrics and view the security assessment in the Model Engineering Engine. Upon gaining insights in the XMANAI platform, they can decide how the results and explanations will be integrated in the actual operations (e.g. through the XMANAI manufacturing apps).

3.5 AI Application Workflow

When the consolidated XAI pipelines have been finalized by the data scientists (in collaboration with data engineers and business users), data engineers are responsible for monitoring their execution. In particular, as depicted in the following figure, the data engineers can access the Execution & Orchestration Engine to view the execution of an XAI pipeline (which they are authorized to view based on the applicable access policies enforced by the Data Access manager). Through the execution logs, they know what happened with each execution of the XAI pipeline so as to troubleshoot any failed executions. In parallel, they are able to monitor the performance metrics (in collaboration with the Model Engineering Engine) and request for scaling the available resources. The data engineers are also responsible for checking that the XAI pipeline results are leveraged by the XMANAI manufacturing apps.

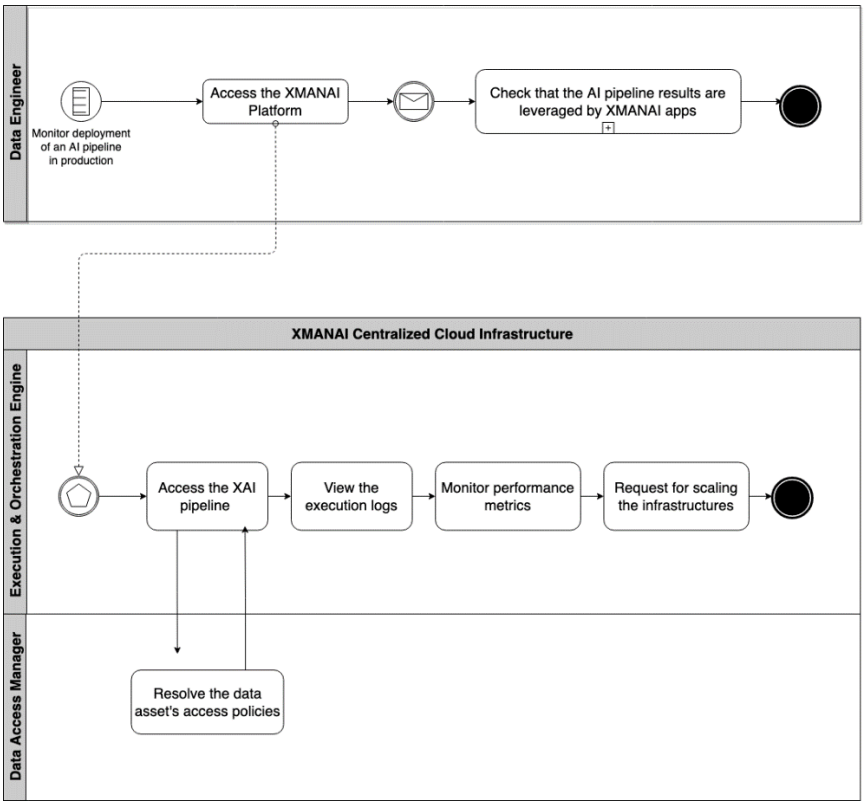

Figure 3-5: AI Application Workflow – Part I

At runtime, the XMANAI manufacturing apps are expected to leverage the deployed XAI pipelines by trigerring the execution of an XAI pipeline that may happen based on schedule or on-demand in the Execution & Orchestration Engine. As depicted in Figure 3-6, once the Execution & Orchestration Engine receives the request to trigger the execution of an XAI pipeline, it checks with the Data Access Manager whether it is authorized to access the specific pipeline's results. If the app is authorized, then the Execution & Orchestration Engine checks how the execution is triggered. If it is triggered based on the predefined schedule of the pipeline, the Execution & Orchestration Engine retrieves the execution results from the Scalable Storage Services (from the centralized cloud or the on-premise installation).

If the pipeline is triggered on-demand, the Execution & Orchestration Engine retrieves the pipeline configuration and executes it in the cloud or on-premise. If the execution has failed, the app receives a failed status response while if the execution is successful, the app receives the execution results.

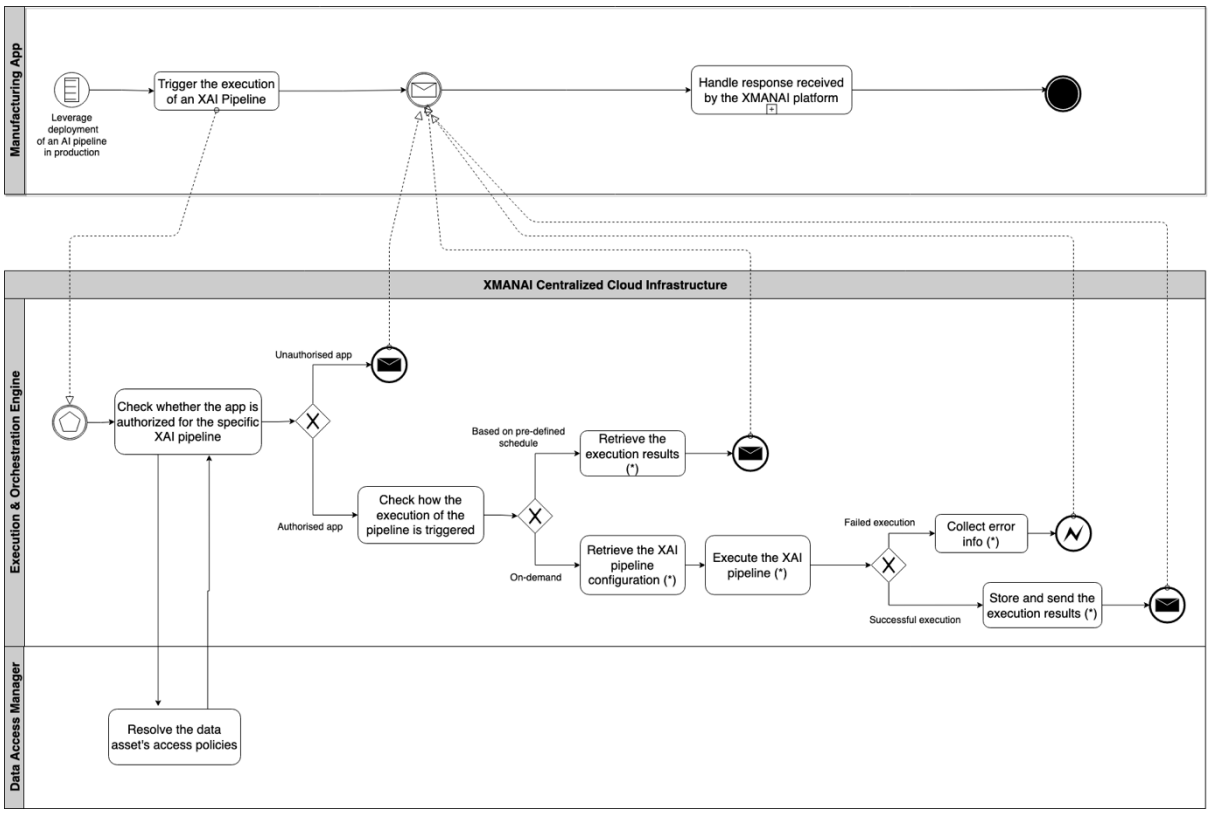

Figure 3-6: AI Application Workflow – Part II

4 XMANAI Platform Components

This section outlines the design of the XMANAI Platform Components in terms of functionalities, mapping to the XMANAI MVP features and technical requirements and interactions. Such components will be further broken down in sub-components in the specifications that accompany the XMANAI Deliverables D2.1 and D3.1. It needs to be clarified that the Data Anonymizer is described only in D2.1 and not in the next paragraphs to avoid content repetition (as the specific component has limited interactions with other components and its functionalities cannot be further broken down).

4.1 Data Access Manager

The Data Access Manager holds a key role within the Platform Management Services Bundle as it provides the required data access control mechanism of the XMANAI platform that regulates the proper and legitimate access to all assets included in the platform. It is available and served through the Core AI Platform in the XMANAI Centralized Cloud deployment.

4.1.1 Overview

The Data Access Manager is undertaking the responsibility of providing the robust and solid access control mechanism that safeguards and prevents the various assets of the XMANAI platform from any unauthorised or unintentional access. This access control mechanism is realised through a flexible and sophisticated **logical access control model** that is formulated by **access control policies**. Access control policies define in an explicit manner the list of operations which are permitted on an asset, by whom and under which conditions or context. The formulated access control policies are effectively combined in order to define the logical access control model which is utilised to formulate the access control decision for each access request. Hence, the access to any underlying asset is regulated by evaluating the condition if the requestor (subject) that aspires to gain access on the specific asset is satisfying the defined access policy (or combination of access policies) for the specific asset or not. If the policy or policies are honoured, the requestor is permitted to access the requested asset else the access is denied.

To meet its goals, the Data Access Manager has to effectively cover many aspects of the access control process. At first, the Data Access Manager facilitates the effective creation of access control policies by the legitimate owners of the assets ensuring that these owners remain always in control of who has access to their assets. Moreover, as the conditions under which access is provided change dynamically based on the usage of the platform or the business scope of each manufacturing organisation, the Data Access Manager enables their update or even deletion with immediate effect, as well as their reuse on other assets if needed. Hence, it covers the complete access policy lifecycle management by undertaking all the required operations and providing the required user interface to the users of the XMANAI platform to perform the access policies definition and management.

The defined access policies are fed to the access control engine in order to formulate the logical access control model which is utilised in the access control decision process. This operation is transparent to the users and it is handled by the Data Access Manager. From this point, the access control mechanism undertakes the regulation of the access to all protected - by access policies - assets by performing the evaluation of each access request against the logical access control model to formulate an access control decision. To achieve this, the Data Access Manager acts as the intermediate component between the components (e.g. from the Data Manipulation Services and AI Execution Services) that aspire to obtain access to any kind of asset of the platform such as projects, datasets, trained models, analytics results and the components that store and manage these assets (e.g. from the Data & Models Governance Services and Scalable Storage Services). This process is facilitated with a set of API endpoints which are provided by the Data Access Manager and leveraged by all components requesting access to any kind of protect asset within the context of the XMANAI platform.

The Data Access Manager consists of: (a) the Policy Engine, and (b) the Policy Editor that are described in detail in the XMANAI Deliverable D2.1.

4.1.2 Mapping to MVP Features & Technical Requirements

The design specifications of the Data Access Manager were driven by the different technical requirements defined in the XMANAI Deliverable D1.2. As depicted in the following table, the relevance of the related requirements is marked as high (if the Data Access Manager is mainly responsible for the specific functionality), medium (if the Data Access Manager contributes to the specific functionality) and low (if the Data Access Manager mostly leverages the specific functionality from other components in the XMANAI architecture).

Table 4-1: Data Access Manager - Mapping to Technical Requirements

No	As a…	I want to …	So that…	Relevance
TR_20	Business user, Data Engineer	be able to store my datasets on my premises	I can only have access to them and I can use them in my analysis	Medium
TR_21	Business user, Data Scientist	able to define the proper access policies on my assets (datasets, AI models, AI pipelines, experiments, analysis results)	I can define who shall have access to my assets and under which circumstances	High
TR_22	Business user, Data Scientist	able to combine multiple access policies on my assets (datasets, AI models, AI pipelines, experiments, analysis results)	I can define more complex access restrictions to my assets	High
TR_23	Business user, Data Scientist	able to define access policies based on various attributes of the requestor or a specific asset (datasets, AI models, AI pipelines, experiments, analysis results)	I can define who shall have access to my assets and under which circumstances	High
TR_24	Business user, Data Scientist	able to update or remove the access policies on my assets (datasets, AI models, AI pipelines, experiments, analysis results)	I can reconsider who shall have access to my assets	High
TR_25	Business user, Data Scientist	able to define the access level of my assets only to my organisation	I can provide access only to my organisation's users	High
TR_26	Business user, Data Scientist	able to define the access level of my assets only to selected users outside my organisation	I can get support from other data scientists	High
TR_27	Business user, Data Scientist	able enforce the access control decision based on my access policies	I can ensure that the access to my assets is always safeguarded	High
TR_28	Data Scientist, Data Engineer, Business User	ensure that only properly authenticated users have access to my assets (datasets, AI models, AI pipelines, experiments, analysis results)	I can ensure their privacy and security	Medium
TR_31	Business user, Data Scientist	ensure that my data are transferred between the different layers of the platform securely	I can ensure that my data will not be disclosed to unauthorized parties	Low

No	As a…	I want to …	So that…	Relevance
TR_46	Business user	see a list of users who has ever had access to an asset that I provided and what activities were performed	I can monitor my assets usage	Medium
TR_48	Business user	XMANAI to register to whom I permitted access to my data or other assets	I can log the sharing my data and other assets with other users	Medium

In addition, the MVP features that are within the scope of the Data Access Manager are depicted in the following table. It needs to be noted that the role of Data Access Manager imposes its involvement in various features in order to provide the required data access control aspects in even if it is not the core component undertaking their implementation. Hence, the bold highlights indicate the MVP features for which the Data Access Manager is primarily responsible.

Table 4-2: Data Access Manager - Mapping to MVP Features

ID	Title
XMANAI_F_UM_001	**Organization-based access (users within the same organization can access all data assets that belong to an organization)**
XMANAI_F_UM_002	**Project-based access (users from different organizations can access data assets that are grouped under a project)**
XMANAI_F_DM_020	**Data Asset Access Policies and Licencing (configuration, management and enforcing)**
XMANAI_F_DM_022	Data Asset Access and Activity Logging
XMANAI_F_DS_028	Data Asset Search and Discovery (including indexing, sorting, filtering, matching level to your asset, view details)
XMANAI_F_DS_029	Secure Transfer of data assets acquired to their legitimate consumers (across platform layers, history log)

4.1.3 Interactions

As described in section 4.1.1, the Data Access Manager constitutes a core component of the XMANAI architecture since it provides the vertical service of the access control mechanism that is leveraged through its exposed APIs by almost all components of the platform through its exposed APIs that require access to the underlying assets of the XMANAI platform. On the other hand, the Data Access Manager interacts with the Identity and Authorisation Manager to retrieve the information related to the requestor of the received access request. To this end, the Data Access Manager mainly interacts via the respective endpoints with the following components:

- The Data Handler for the definition of the access policies of a newly introduced asset or providing the list of assets that can be legitimately exported.
- The Data Manipulation Engine for providing the list of datasets that can be accessed in the data manipulation tasks.
- The Execution & Orchestration Engine for providing the list of XAI pipelines that can be legitimately executed.
- The Identity and Authorisation Manager for the retrieval of the requestor's identity information.
- The Provenance Engine for providing the data asset access logging information.
- The XAI Marketplace for providing the list of accessible assets in the marketplace and for checking the existence of an active contract for an asset.

- The XAI Pipeline Manager for the definition of the access policies of XAI pipelines.
- The XAI Visualisation Engine for providing the list of datasets that can be accessed in the various visualisation tasks.

It should be noted that the Data Access Manager interacts with almost all the components of the platform in order to regulate all the access to the various assets and the list above as is not an exhaustive list. For simplicity reasons, all the main interactions as listed above are depicted in the following diagram.

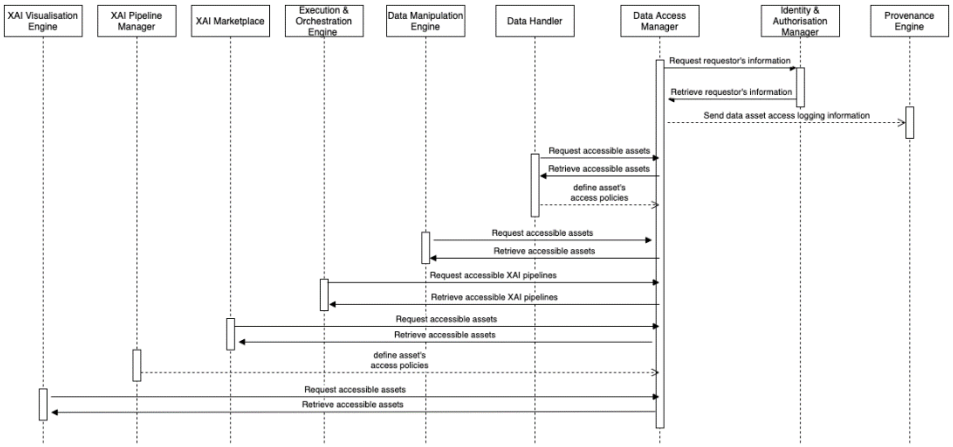

Figure 4-1: Data Access Manager – Interactions with other XMANAI Components

4.2 Data Handler

The Data Handler is one of the key components of the XMANAI Platform belonging to its Data Collection and Governance Services bundle. It is responsible for the secure collection and management of data assets in the XMANAI Platform.

4.2.1 Overview

The Data Handler provides the basic services for the secure collection, storage and management of data and other relevant artefacts in the XMANAI platform. The data collection is one of the main functionalities of the Data Handler. The component provides a full set of possible data collection functionalities:

- Pro-active collection of data from external APIs according to the configurable time plan and using API specific configurable connectors, transformation of the collected data to the XMANAI data model(s) and propagation of the data in the XMANAI data store. The project will develop a set of connectors necessary for implementation of the planned XMANAI demonstrators and will provide documentation for including additional connectors in XMANAI.
- Provision of an API enabling third party applications registered in XMANAI to submit to the Data Handler, data conformed to the XMANAI data model or any files.
- Upload of data or other files via the user interface.

Another important functionality of the Data Handler is the management of data projects. A data project in XMANAI represents a secure storage area in which users, members of an organization that belong to the same data project, can collect and manage the associated to the project datasets, trained models, results of computations and other relevant files. Using the Data Handler's GUI the users will manage their projects and associated with them artefacts.

The above-mentioned functionalities are based on the main functionality provided by the Data Handler – storage and management of data and other artefacts. The Data Handler provides an API enabling other components to save or get data or other assets from the XMANAI data store(s). The Data Handler includes the API Data Harvester, the File Data Harvester, the Data Gateway, the File/Data Manager and the Data Exporter components. The detailed architecture of the Data Handler and functionalities of underlying components are presented in the XMANAI Deliverable D2.1.

4.2.2 Mapping to MVP Features & Technical Requirements

The design of the Data Handler addresses the technical requirements presented in the XMANAI Deliverable D1.2. The following table presents the subset of them, which are relevant for the Data Handler. The relevance of the related requirements is marked as high (if the Data Handler is mainly responsible for the specific functionality), medium (if the Data Handler contributes to the specific functionality) and low (if the Data Handler mostly leverages the specific functionality from other components in the XMANAI architecture).

Table 4-3: Data Handler - Mapping to Technical Requirements

No	Task No	As a…	I want to …	So that…	Relevance
TR_1	T2.1	Data Scientist, Data Engineer, Business User	add/edit/remove new data sources for data/metadata import	Data/metadata from these sources can be available for analysis	High
TR_2	T2.1	Data Scientist, Data Engineer, Business User	define rules how often the data have to be collected	I can get regular data updates from an external source	High
TR_3	T2.1	Data Scientist, Data Engineer, Business User	upload data as single file or batch of files	I can work with these data in the platform	High
TR_4	T2.1	Data Engineer	Delete data and its associated metadata	I can remove data, which I don't need anymore	High
TR_12	T2.1	Data Scientist, Data Engineer, Business User	add/edit metadata for my data (based on a common metadata model)	I can improve the quality of data for further reuse	Medium
TR_13	T2.1	Data Scientist, Data Engineer, Business User	define how my data are mapped to a data model	the data types and semantics of my data can be available to anyone who uses my data	High
TR_14	T2.1	Data Scientist, Data Engineer, Business User	define whether and how my data should be cleaned	I can increase their quality before they are stored	Low
TR_15	T2.1	Data Scientist, Data Engineer, Business User	manage (create new, rename, move, delete) my datasets	I can work with them in the platform	High
TR_16	T2.1	Business user, Data Scientist, Data Engineer	update existing datasets with new data	I can perform a new analysis	High

No	Task No	As a…	I want to …	So that…	Relevance
TR_17	T2.1 & T2.3	Data Scientist	download data that I have legitimate access as a file	I can use them offline	High
TR_18	T2.1 & T2.3, T5.X	Data Scientist	retrieve data that I have legitimate access through an API	I can use them in the XMANAI manufacturing apps	High
TR_19	T2.1 & T2.3, T3.2, T3.3	Data Scientist, Data Engineer	export data samples from the dataset	I can view them in other XMANAI components and external tools	High
TR_20	T2.1, T2.2, T3.5	Business user, Data Engineer	be able to store my datasets on my premises	I can only have access to them and I can use them in my analysis	High

The Data Handler is responsible or contributes to the provision of several MVP features. The table below illustrates the situation. The MVP features for which the Data Handler is primarily responsible are marked in bold.

Table 4-4: Data Handler - Mapping to MVP Features

ID	Title
XMANAI_F_DI_004	**Data Sources Management (add/edit/remove/configure/schedule)**
XMANAI_F_DI_005	**Data Secure Uploading as file(s)**
XMANAI_F_DI_006	**Data Secure Uploading via API**
XMANAI_F_DI_007	Data Mapping to a Data Model and Harmonization (data type casting, transformation to common measurement unit/timestamp)
XMANAI_F_DI_009	Data Cleansing (quality checks, cleaning rules definition and execution before storage)
XMANAI_F_DI_011	**Data Storage in central XMANAI Cloud Storage**
XMANAI_F_DI_012	**Data Storage on-premise / in private cloud**
XMANAI_F_DM_014	Dataset Management (add/edit/remove asset and metadata, versioning)
XMANAI_F_DM_016	Results Management (edit/remove asset and metadata, versioning)
XMANAI_F_DM_017	AI Model Management (create/store/update/delete/clone/export/import, configure, versioning, register/import)
XMANAI_F_DM_018	AI Pipeline Management (create/store/update/delete/clone/export/join, view IPR of assets involved, define templates, versioning)
XMANAI_F_DM_019	**Data Asset Export (download file or via API depending on IPR)**

4.2.3 Interactions

The Data Handler is one of the basic components of the XMANAI architecture providing the functionality of collecting and managing of data-related artefacts. For this purpose it provides an API, which is used by the components of the higher levels in the XMANAI architecture. In these interactions the Data Handler plays a passive role replying to the requests of other components. Proactively the Data Handler interacts with:

- The Data Access Manager to validate the permissions of the requests coming from the users or other components.

- The Provenance Engine to update the provenance information about data and other XMANAI assets.

The figure below illustrates the interaction of the Data Handler with other components on the example of the data harvesting process. It needs to be noted that the interactions of the Data Handler with the Data Access Manager, and the Identity & Authorisation Manager, are not currently depicted in Figure 4-2.

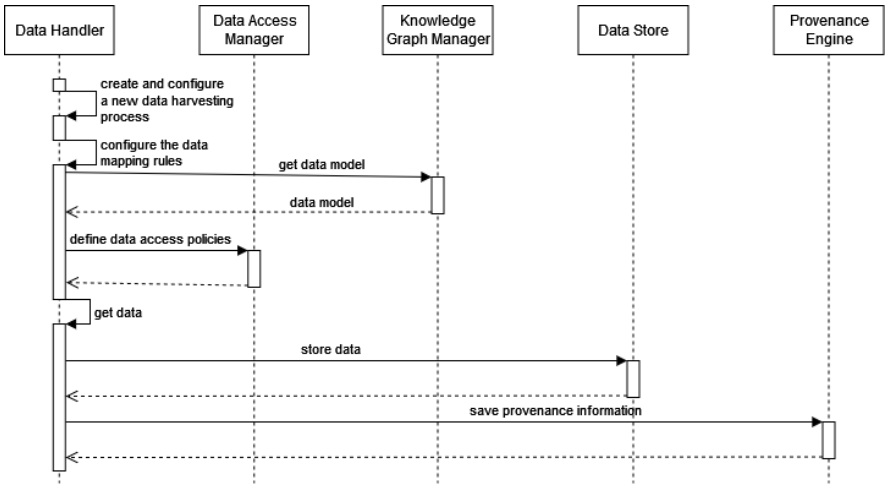

Figure 4-2: Data Handler – Interactions with other XMANAI Components

4.3 Data Manipulation Engine

The Data Manipulation Engine is an important part of the Data Manipulation Services Bundle as it provides functionalities for data pre-processing that enable enhanced data understanding and allow the transformation of the input data in formats appropriate for other analysis purposes, including visualisations and applications of artificial intelligence models. It is leveraged by data scientists and data engineers, but also by business users, as it provides a wide range of data manipulation functionalities, some applicable only by users with a data science background and others straightforward even for non-technical users. It is available and served through the Core AI Platform in the XMANAI Centralized Cloud deployment and in the On-Premise XMANAI deployments.

4.3.1 Overview

The Data Manipulation Engine is the component responsible for managing all the operations related to the transformations of the stored data assets before these are utilised as input in visualisations and/or machine learning models. As previously mentioned, these transformations may aim to increase the users' understanding of the underlying data or to make the data suitable for usage in other operations, such as visualisations and AI models, notwithstanding the possibility of simply creating derivative data foreseen to be useful to other stakeholders. The scope of the functionalities provided by the component spans across two axes:

- Provision of configurable data preparation templates for guided data pre-processing and filtering functionalities, which can be directly leveraged in the production settings upon configuration.
- Provision of low-level and thus more flexible data preparation functionalities to allow a less constrained experimentation during (mainly) the data scientists' data exploration and intuition development phase.

The actual execution of the aforementioned data preparation functions is performed in collaboration with the Execution & Orchestration Engine (described in Section 4.4). The following non-exhaustive

list of functionalities offered by the Data Manipulation Engine provides more concrete insights regarding its scope:

- Computation of summary statistics of each field included in the retrieved datasets (mean, median, standard deviation etc.), broader investigation of the fields' distributions and correlation checking among pairs of fields.
- Application of dimensionality reduction and data augmentation techniques.
- Provision of simple and advanced mathematical operations, including window and cumulative aggregations.
- Definition and application of filters to allow static or dynamic selection of appropriate data subsets.
- Data cleaning and harmonisation functionalities, including missing values' imputation and value replacement options.
- Commonly used feature preparation functionalities, ranging from information extraction from datetime fields (e.g., day of week, hour, etc.) to scaling and encoding values of selected columns to be ingested by AI models.
- Datasets' combinations (either through joining or appending them) and column operations, ranging from simple ones like deletion and duplication to more advanced operations, e.g. conditional creation of new columns based on existing values in other columns.

The operations performed by the Data Manipulation Engine can be leveraged both independently, but also through their inclusion in more complex XMANAI AI pipelines, thus allowing their seamless combination with other data analysis steps, e.g., those provided by the Model Engineering Engine. The mechanism that enables the extended pipeline creation and management is provided by the XAI Pipeline Manager (described in Section 4.11).

The Data Manipulation Engine consists of: (a) the Data Preparation Engine, and (b) the Interactive Data Exploration and Experimentation Tool (partially), that are described in detail in the XMANAI Deliverable D3.1.

4.3.2 Mapping to MVP Features & Technical Requirements

The Data Manipulation Engine is designed taking into consideration different technical requirements defined in the XMANAI Deliverable D1.2. As depicted in the following table, the relevance of the related requirements is marked as high (if the Data Manipulation Engine is mainly responsible for the specific functionality), medium (if the Data Manipulation Engine contributes to the specific functionality) and low (if the Data Manipulation Engine mostly leverages the specific functionality from other components in the XMANAI architecture).

Table 4-5: Data Manipulation Engine - Mapping to Technical Requirements

No	As a…	I want to …	So that…	Relevance
TR_34	Business User, Data Scientist, Data Engineer	search and explore other data/AI assets on an in a user-friendly way (e.g. based on metadata, with sorting, filtering, matching level)	I can easily find what I am looking for	Low
TR_35	Data Scientist, Data Engineer	view metadata of the selected asset (e.g. datasets, AI models, AI pipelines, analysis results)	I can determine if the asset addresses my needs	Low
TR_41	Data Scientist	consistently handle missing data in my data or by finding other relevant datasets	I can anticipate data drift	Medium

No	As a…	I want to ...	So that…	Relevance
TR_50	Data Scientist	query data to which I have legitimate access	I can find a subset of the data I can use in my analysis	Medium
TR_51	Data Scientist	know the data types and the semantics per field that appears in the data	I can quickly understand the data that I will use in an analysis	Low
TR_52	Data Scientist	preview a sample of the data	I can obtain a more concrete understanding of the data at hand	High
TR_53	Data Scientist	view data distribution/profiling charts or summary statistics for the data (e.g. number of missing values, min/max values)	I can monitor data drifting issues	High
TR_55	Data Scientist	create new features based on the current data (like min, max, mean values) that will be part of the same dataset	I can have more informative datasets, depending on the task	High
TR_56	Data Scientist	handle missing values (impute)	I can prepare the data for the subsequent AI analysis	High
TR_57	Data Scientist	encode categorical data	I can prepare the data for the subsequent AI analysis	Medium
TR_58	Data Scientist	apply scaling and data normalization	I can prepare the data for the subsequent AI analysis	Medium
TR_59	Data Scientist	easily split the data for training and evaluation (data segmentation)	I can train and apply the models as I see fit	Medium
TR_60	Data Scientist	apply simple transformations on the data	I make them more appropriate for processing and visualisation	High
TR_61	Data Scientist	change the data type of some features	I can manipulate them according to the needs of an AI model (e.g. convert to datetime)	High
TR_62	Data Scientist	apply data augmentation techniques	I can reduce overfitting of the models	High
TR_63	Business User	perform calculations over my data	I can keep track of important Key Performance Indicators that are important for my business	High
TR_70	Data Scientist, Data Engineer	define pipelines that can be used as templates for specific problems	my colleagues and I can reuse them	Low
TR_79	Data Scientist	define summary statistics to be computed for an AI pipeline or part of it	I can explain the behavior of the inputs and / or outputs of a pipeline / model	High
TR_93	Data Scientist	have a common metadata model for describing my AI pipelines, experiments, results	I can share them with other users	Low

In addition, the MVP features that are within the scope of the Data Manipulation Engine are depicted in the following table. It needs to be noted that the bold highlights indicate the MVP features for which the Data Manipulation Engine is primarily responsible.

Table 4-6: Data Manipulation Engine - Mapping to MVP features

ID	Title
XMANAI_F_DI_007	Data Mapping to a Data Model and Harmonization (data type casting, transformation to common measurement unit/timestamp)
XMANAI_F_DM_014	Dataset Management (add/edit/remove asset and metadata, versioning)
XMANAI_F_DM_016	Results Management (edit/remove asset and metadata, versioning)
XMANAI_F_DM_018	AI Pipeline Management (create/store/update/delete/clone/export/join, view IPR of assets involved, define templates, versioning)
XMANAI_F_DS_028	Data Asset Search and Discovery (including indexing, sorting, filtering, matching level to your asset, view details)
XMANAI_F_DP_030	**Data View & Visualisation (query data, view distribution, statistics, aggregations, time periods, descriptive analytics, preview sample)**
XMANAI_F_DP_031	**Data Manipulation (merge, split, augment, resample, aggregate, create new features, handle missing values, etc.)**
XMANAI_F_DP_032	**Data Transformation (normalisation, encoding, modifying data types, etc.)**
XMANAI_F_AI_035	AI Pipeline Design (define, configure, register AI model, add annotations/comments, reuse common features)
XMANAI_F_AI_036	AI Pipeline Execution & Evaluation on XMANAI Common Cloud (experimentation vs production, run automatic tests for AI models, run scheduling, configuration)
XMANAI_F_AI_037	AI Pipeline Execution & Evaluation on Premise / Private Cloud (experimentation vs production, run automatic tests for AI models, run scheduling, configuration)
XMANAI_F_AI_039	AI Pipeline Results Management (store, export, add summary statistics, easily compare with real values (for predictions), retrieve via API)
XMANAI_F_AI_040	AI Pipeline Results Visualisation (configuration of various charts, add comments, store, export, run on cloud vs on premise)

4.3.3 Interactions

In terms of expected interactions with other XMANAI components, the Data Manipulation Engine has (or leverages) interfaces with the following components:

- The Data Access Manager indirectly for ensuring that access to data is performed according to the user defined policies.
- The Knowledge Graph Manager through which it retrieves stored data along with accompanying information from the data model. The Data Manipulation Engine also interacts with the Knowledge Graph Manager to handle the way processed data can be properly stored and made available for usage by this or other components.
- The Execution & Orchestration Engine both directly or indirectly depending on whether the component's functionalities are integrated in an AI pipeline or the direct execution invocation functionalities are used instead.
- The Identity & Authorisation Manager, similar to the Data Access Manager, is indirectly connected to the Data Manipulation Engine to ensure all users can smoothly perform the operations foreseen for them and for their organisations.
- The Model Engineering Engine may act as a consumer of the results of a sequence of data manipulation tasks.
- The Provenance Engine for tracking the lineage of the data processed by the Engine.
- The XAI Pipeline Manager for handling the inclusion of data manipulation tasks in the overall AI pipelines, a process that includes numerous interactions, indicatively including the retrieval

of data manipulation tasks' configuration, the invocation of the execution services for the respective tasks, the retrieval and exchange of the corresponding logs etc.

- The XAI Visualisation Engine, either directly or indirectly, again depending on the mode that is utilized, i.e., through an AI pipeline or independently. Many of the execution results of the configured and executed data manipulation functionalities are best consumed in a visual way, thus interaction with the Visualization Engine to configure the appropriate charts is foreseen.

An indicative view of the expected interfaces is depicted in the following diagram. It needs to be noted that the interactions of the Data Manipulation Engine with the Data Access Manager, and the Identity & Authorisation Manager, are not currently depicted in Figure 4-3.

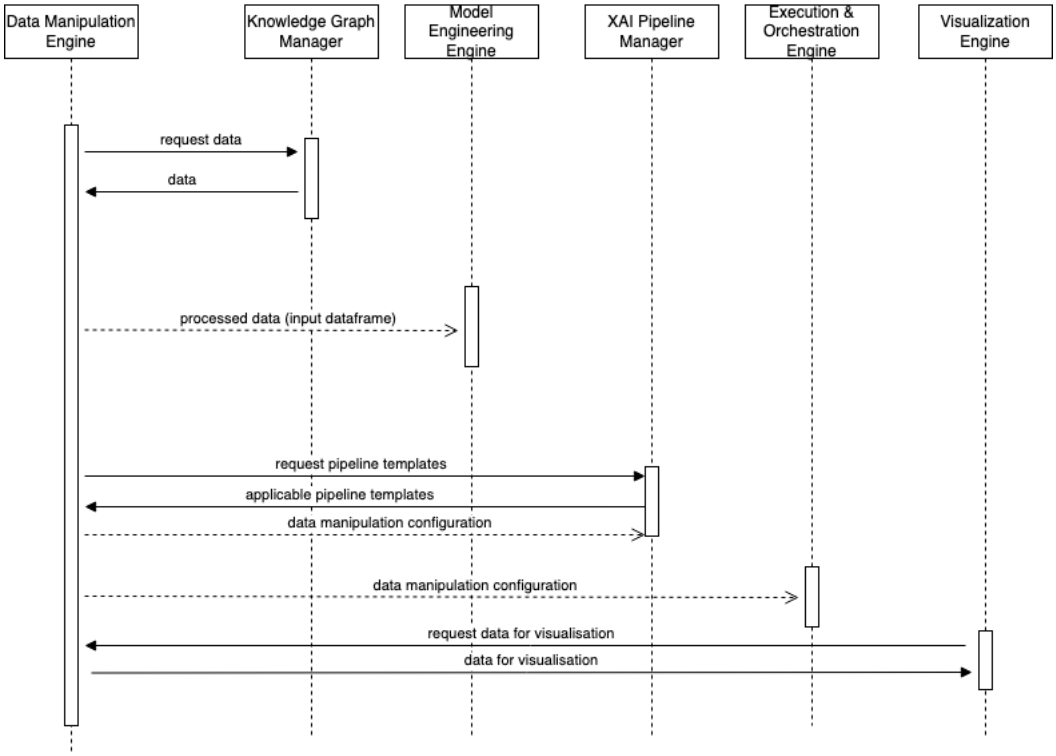

Figure 4-3: Data Manipulation Engine – Interactions with other XMANAI Components

4.4 Execution & Orchestration Engine

The Execution & Orchestration Engine is responsible for the execution of the fully configured XAI Pipelines created by the XAI Pipeline Manager beforehand and belongs to the AI Execution Services Bundle that is available through the Centralized Cloud and the On-Premise (Private Cloud) deployments of the XMANAI Platform.

4.4.1 Overview

From an adaptability point of view, the Execution & Orchestration Engine offers an elastic execution environment, allowing the orchestration of pipelines belonging to distinct execution classes and presenting different requirements in order to carry out the corresponding tasks:

- **Batch environments** that need high/large volumes of data to be collected and processed in groups. This type of execution is mainly suited for historical, batch/ static data that are not characterized by continuous flows of information, resulting in a generally slower execution process over wide datasets.

- **Near real time environments**, on the other hand, are best suited for smaller amounts of data that are continuously produced and, as a result, require lighter computations to be performed in a repetitive way to keep up with the steady production of new data samples.

The Execution & Orchestration Engine is responsible for triggering the execution of AI pipelines based on specific schedule or on-demand. Details for all executions are stored and become available to the execution log.

Finally, this component offers a scalable and optimized architecture that can grow and shrink depending on the tasks demand for resources. It also accounts for parallelism, allowing multiple tasks to be executed at the same time, if those are not dependant to each other.

The Execution & Orchestration Engine requires as input the pre-assembled pipeline to be executed, as well as the scheduling settings required in order to properly execute the pipeline at the right time or when the predefined set of constraints is satisfied. Both of these settings are served to the Execution & Orchestration Engine by the XAI Pipeline Manager and, in particular, by the Pipeline Serving & Monitoring Engine sub-component.

The expected output from the Execution & Orchestration Engine comes under different forms and depends on the pipeline type and the execution steps. This component basically outputs a non-standard result that may range from a dataframe (that encapsulates a set of predictions and explanations) to a trained model, and is predefined by the user who assembled the pipeline.

4.4.2 Mapping to MVP Features & Technical Requirements

The Execution & Orchestration Engine is designed taking into consideration different technical requirements defined in the XMANAI Deliverable D1.2. As depicted in the following table, the relevance of the related requirements is marked as high (if the Execution & Orchestration Engine is mainly responsible for the specific functionality), medium (if the Execution & Orchestration Engine contributes to the specific functionality) and low (if the Execution & Orchestration Engine mostly leverages the specific functionality from other components in the XMANAI architecture).

Table 4-7: Execution & Orchestration Engine - Mapping to Technical Requirements

No	As a…	I want to …	So that…	Relevance
TR_70	Data Scientist, Data Engineer	define pipelines that can be used as templates for specific problems	my colleagues and I can reuse them	Low
TR_71	Data Scientist, Data Engineer	clone a designed pipeline	create alternative version of the pipeline and improve its performance without re-creating it from scratch	Low
TR_73	Data Scientist, Data Engineer	execute step-by-step an AI pipeline over sample data	ensure that the result is the intended one	High
TR_74	Data Scientist, Data Engineer	reuse common features in different pipelines	do not need to recompute them	Medium
TR_80	Data Scientist	add annotations and comments in AI pipelines	better explain the results	Low
TR_86	Data Scientist	run automatic tests on the registered AI models (when including them in an AI pipeline)	quickly check that the trained models are robust and fault-tolerant	Medium
TR_103	Data Scientist, Data Engineer	keep a history of experiments performed (pipeline runs)	compare results & improve the pipeline	Low
TR_104	Data engineer	schedule the execution of the AI pipelines	have at my disposal up-to-date results	Medium

No	As a…	I want to …	So that…	Relevance
TR_105	Business User, Data Scientist	store the execution results	retrieve them to use them in external systems	High
TR_106	Data Engineer	receive notifications when certain metrics exceed defined thresholds	timely investigate problems in deployed pipeline	Low
TR_107	Data Scientist, Data Engineer	have the required resources and automatic parallelization, when dealing with big data manipulation	execute an AI pipeline in a faster and more reliable way	Medium
TR_108	Business user, Data Scientist	execute AI pipelines locally on my premise environment	I can perform my analysis in a secure and trusted environment	High
TR_112	Data Engineer	containerize the developed AI pipelines	they can be more easily deployed	Medium
TR_129	Data Scientist	compare performances of different AI pipelines	track the improvements of the different versions of my pipelines / models	Medium

In addition, the MVP features that are within the scope of the Execution & Orchestration Engine are depicted in the following table. It needs to be noted that the bold highlights indicate the MVP features for which the Execution & Orchestration Engine is primarily responsible.

Table 4-8: Execution & Orchestration Engine - Mapping to MVP features

ID	Title
XMANAI_F_DM_018	AI Pipeline Management (create/store/update/delete/clone/export/join, view IPR of assets involved, define templates, versioning)
XMANAI_F_DM_024	AI Pipeline Security Assessment
XMANAI_F_AI_035	AI Pipeline Design (define, configure, register AI model, add annotations/comments, reuse common features)
XMANAI_F_AI_036	**AI Pipeline Execution & Evaluation on XMANAI Common Cloud (experimentation vs production, run automatic tests for AI models, run scheduling, configuration)**
XMANAI_F_AI_037	**AI Pipeline Execution & Evaluation on Premise / Private Cloud (experimentation vs production, run automatic tests for AI models, run scheduling, configuration)**
XMANAI_F_AI_039	AI Pipeline Results Management (store, export, add summary statistics, easily compare with real values (for predictions), retrieve via API)

4.4.3 Interactions

The Execution & Orchestration Engine mainly interacts with:

- **XAI Pipeline Manager** from which it obtains the pipeline configuration to be executed and the scheduling information to orchestrate the pipeline execution at the appropriate time or when the predefined requirements are met by the application environment. The Execution & Orchestration Engine also returns back to the XAI Pipeline Manager the logging data and monitoring information generated during the pipeline execution, allowing the XAI Pipeline Manager to properly display such information.
- **XAI Visualization Engine** to which the execution results that are generated by the pipeline are provided. Those may come in different formats, depending on the pipeline architecture and objective of the execution. The results shared with the XAI Visualization Engine are transferred in a raw format, without performing any cleaning, conversion, or processing, as those operations are outside the scope of the Execution & Orchestration Engine.

The above interactions of the Execution & Orchestration Engine are depicted in Figure 4-4. It needs to be noted that the Execution & Orchestration Engine also interacts with the Data Access Manager, and

the Identity & Authorisation Manager for authentication and authorization, even though the specific interactions are not currently depicted.

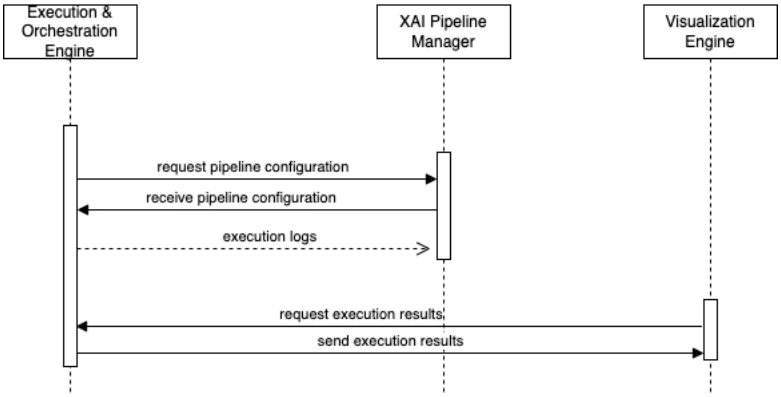

Figure 4-4: Execution & Orchestration Engine – Interactions with other XMANAI Components

4.5 Identity and Authorisation Manager

The Identity and Authorisation Manager constitutes a key component of the Platform Management Services Bundle undertaking all the operations related to user management and the authorised intercommunication between the various layers or components of the platform. It is available and served through the Core AI Platform in the XMANAI Centralized Cloud deployment.

4.5.1 Overview

The Identity and Authorisation Manager provides the required user management operations of the XMANAI platform and is responsible for the complete user management lifecycle and the authentication mechanism of the platform. Effective and efficient user management constitutes the cornerstone of each platform as it facilitates all the internal operations of the platform across the different services as well as the access control operations for the assets of the platform. The Identity and Authorisation Manager undertakes the responsibility of providing the single core identity provider of the platform as well as the platform's strong and robust authentication mechanism built on top of this identity provider.

Within the context of XMANAI, based on the requirements elicited in D1.2 and D6.1, the users are classified into organisations in order to facilitate the easy and flexible sharing of assets and operations among the users belonging under the same organisation. To this end, all the users under the same organisation share the same rights on the organisation's assets and operations such as projects, datasets, trained models, analytics results with the exception of top-level users which have more escalated rights such as the initial registration of the organisation in the platform, the invitation and management of users under their organisation, as well as the signing of smart contracts through the XAI Marketplace. The Identity and Authorisation Manager provides all the operations related to the user management operations through an easy-to-use user interface while also providing all the related background operations which are hidden to the users such as retrieval of user's or the organisation's information in order to be used in the access control process. Furthermore, the Identity and Authorisation Manager provides the strong authentication mechanism of the XMANAI platform that imposes strict restrictions to the access of all its offerings and services for the registered and successfully logged in users only. To this end, the Identity and Authorisation Manager interacts mainly with the Data Access Manager in order to facilitate the access control operations while also providing the user's or the organisation's information to the components that facilitate the exchange of data assets between the different registered organisations of the platform, such as the XAI Marketplace.

In addition to the user management operations, the Identity and Authorisation Manager regulates the intercommunication between the various layers or components towards the enhanced security and integrity of the exchanged information and utilised resources within the scope of the platform. In

particular, the Identity and Authorisation Manager provides the authorisation mechanism that controls which components can intercommunicate via their respective services or exposed endpoint in order to ensure that unauthorised access will not be granted by intercepting all the intercommunication between the components of the platform. To achieve this, the Identity and Authorisation Manager interacts with the Data Access Manager in order to exchange the required information for the formulation of the authorisation decision.

The design specifications of the Identity and Authorisation Manager are described in detail in the XMANAI Deliverable D2.1.

4.5.2 Mapping to MVP Features & Technical Requirements

The design specifications of the Identity and Authorisation Manager were driven by the different technical requirements defined in the XMANAI Deliverable D1.2. As depicted in the following table, the relevance of the related requirements is marked as high (if the Identity and Authorisation Manager is mainly responsible for the specific functionality) and medium (if the Identity and Authorisation Manager contributes to the specific functionality).

Table 4-9: Identity and Authorisation Manager - Mapping to Technical Requirements

No	As a…	I want to ...	So that…	Relevance
TR_25	Business user, Data Scientist	able to define the access level of my assets only to my organisation	I can provide access only to my organisation's users	Medium
TR_26	Business user, Data Scientist	able to define the access level of my assets only to selected users outside my organisation	I can get support from other data scientists	Medium
TR_28	Data Scientist, Data Engineer, Business User	ensure that only properly authenticated users have access to my assets (datasets, AI models, AI pipelines, experiments, analysis results)	I can ensure their privacy and security	High
TR_31	Business user, Data Scientist	ensure that my data are transferred between the different layers of the platform securely	I can ensure that my data will not be disclosed to unauthorized parties	High
TR_46	Business user	see a list of users who has ever had access to an asset that I provided and what activities were performed	I can monitor my assets usage	Medium
TR_48	Business user	XMANAI to register to whom I permitted access to my data or other assets	I can log the sharing my data and other assets with other users	Medium

In addition, the MVP features that are within the scope of the Identity and Authorisation Manager are depicted in the following table. It needs to be noted that the bold highlights indicate the MVP features for which the Identity and Authorisation Manager is primarily responsible.

Table 4-10: Identity and Authorisation Manager - Mapping to MVP Features

ID	Title
XMANAI_F_UM_001	**Organization-based access (users within the same organization can access all data assets that belong to an organization)**
XMANAI_F_UM_002	**Project-based access (users from different organizations can access data assets that are grouped under a project)**
XMANAI_F_DM_020	Data Asset Access Policies and Licencing (configuration, management and enforcing)

XMANAI_F_DM_021	**Data Asset Secure Transfer (through platform layers, operation and enforcing)**
XMANAI_F_DM_022	Data Asset Access and Activity Logging
XMANAI_F_DS_029	Secure Transfer of data assets acquired to their legitimate consumers (across platform layers, history log)

4.5.3 Interactions

As described in section 4.5.1, the Identity and Authorisation Manager provides a vertical service of the XMANAI architecture that provides the single identity provider and the authorisation mechanism for the intercommunication of all the components. To this end, the Identity and Authorisation Manager interacts with the users of the platform as well as with the components of the platform as follows:

- The user of the platform in order to facilitate the registration, invitation and login processes.
- All components in order to provide the identity information for the requested user.
- All components in order to verify and authorise the intercommunication with a specific component.

For simplicity reasons, all the main interactions as listed above are depicted in the diagram referring to Component X that applies to all components of the architecture.

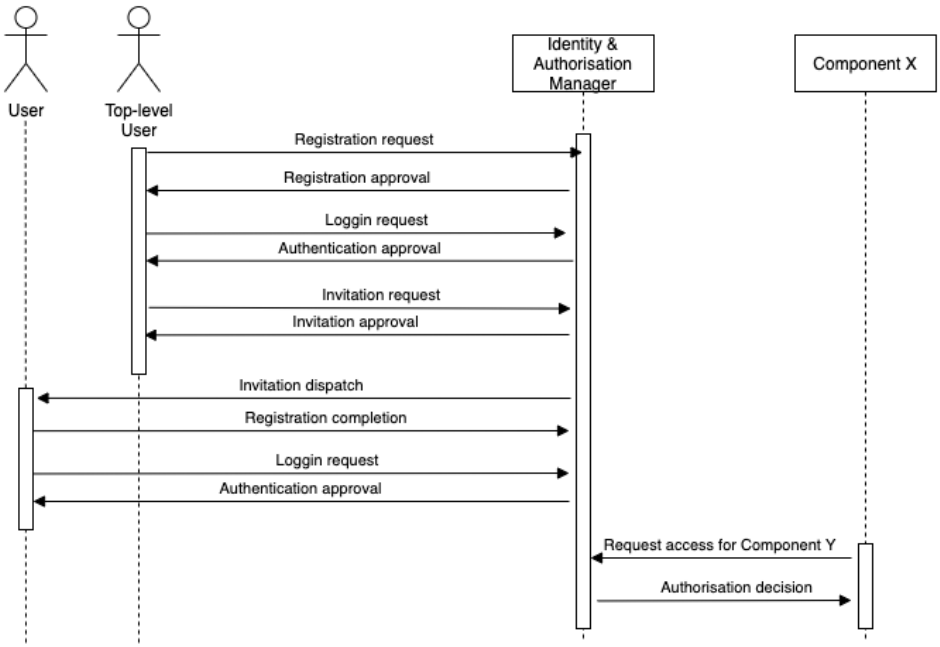

Figure 4-5: Identity and Authorisation Manager – Interactions with other XMANAI Components

4.6 Knowledge Graph Manager

The Knowledge Graph Manager is a component of the XMANAI Data Manipulation Services Bundle which provides the other XMANAI components with the ability to legitimate and facilitate data assets access to the design and execution of AI pipelines. It is also responsible (together with the Data Handler) for providing data explainability for Data Scientist and Data Engineers. It is available and served through the Core AI Platform in XMANAI Centralized Cloud deployment and its data extraction functionality is also available through the on-premise XMANAI deployments.

4.6.1 Overview

The Knowledge Graph Manager (KGM) aims at helping to provide semantics, vocabularies and generally information about the structure of input data, with the goal of increasing the data explainability while supporting the maintenance of the XMANAI Data Models. KGM makes ingested data transparent for Data Scientists and Data Engineers. This component visualizes the XMANAI Data Model, both in a tabular and graph view, to facilitate understanding of its contents as well as to allow the modification of the data structures if necessary.

The business user takes advantage of the Data Model Visualisation Dashboard to step by step find its desired structure and explanation about imported data (the imported data will be mapped to the Data Model by Data Handler Component). A sample of the data is also prepared for preview by the Knowledge Graph Manager in order to provide more tangible insights. Such indirect explanations provide useful information (clarity) for the Data scientists and makes the data understandable, consequently he/she can easily use the data for creating AI pipelines.

Managing the model lifecycle is one of the main functionalities of the Knowledge Graph Manager to update and improve the XMANAI Data Model. This functionality makes the Data Model expandable. This enables the system administrator (Admin) with special access to the XMANAI Data Model storage, to enrich and update the data model. The admin analyses the Data Model entities proposals made by users in order to enlarge the Model.

One other main functionality of the Knowledge Graph Manager is to facilitate access to the legitimate data assets that were previously ingested in the XMANAI platform. This data extract can be used to design and execute several AI pipelines by Data Scientists and Data Engineers.

For detailed information on the Knowledge Graph Manager, please refer to XMANAI Deliverable D3.1.

4.6.2 Mapping to MVP Features & Technical Requirements

The Knowledge Graph Manager is based on different technical requirements defined in the XMANAI Deliverable D1.2. Table 4-11 addresses these Technical requirements. This table is extended by a column named “Relevance” that represents the technical requirements relevance in 2 levels as follows: high (if the Knowledge Graph Manager is mainly responsible for the specific functionality) and medium (if the Knowledge Graph Manager contributes to the specific functionality).

Table 4-11: Knowledge Graph Manager - Mapping to Technical Requirements

No	As a …	I want to …	So that…	Relevance
TR_6	Data Scientist, Business User	search for concepts, fields and relationships to an existing data model	I can ensure my specific data needs are addressed	High
TR_7	Admin, Data Engineer	manually add new concepts, fields and relationships to an existing data model	I can ensure the needs of data scientists and business users are addressed	High
TR_8	Admin, Data Engineer	manually update the concepts, fields and relationships of an existing data model	I can change the model over time	High
TR_9	Admin, Data Engineer	manually control the different versions of the data model	I can view and retrieve all the history of edits	High
TR_10	Admin, Data Scientist	view the different data models	I can use an appropriate model per domain	High
TR_11	Data Scientist	generate different representations (e.g. graph) of the data model	I can make it available to other applications / components	High

No	As a …	I want to …	So that…	Relevance
TR_13	Data Scientist, Data Engineer, Business User	define how my data are mapped to a data model	the data types and semantics of my data can be available to anyone who uses my data	Medium
TR_19	Data Engineer, Data Scientist, Business User	preview a sample of the data	I can obtain a more concrete understanding of the data at hand	High

The following table includes the MVP features that are in the scope of the Knowledge Graph Manager. **XMANAI_F_DI_008** feature that is highlighted in bold is the feature that the Knowledge Graph Manager is directly responsible for.

Table 4-12: Knowledge Graph Manager - Mapping to MVP Features

ID	Title
XMANAI_F_DI_007	Data Mapping to a Data Model and Harmonization (data type casting, transformation to common measurement unit/timestamp)
XMANAI_F_DI_008	**Data Model Management (view model, add/edit/remove model concepts, versioning different representations)**
XMANAI_F_DM_019	Data Asset Export

4.6.3 Interactions

As explained in section 4.6.1, the Knowledge Graph Manager is responsible for explainabity and transparency of ingested data by providing the domain semantics and vocabularies through the XMANAI Data Model. The input data is mapped to the XMANAI Data Model by the Data Handler component. To achieve this, the Data Handler component needs to interact with the Data Model through the Knowledge Graph Manager.

The Knowledge Graph Manager also provides facility of enriched data extraction from storage to other XMANAI components. For this reason, it has interactions with different XMANAI components including the Data Manipulation Engine, the Model Engineering Engine and the Orchestration & Execution Engine to provide data with information from the data model. In terms of expected interactions with other XMANAI components, it interfaces (or leverages) with the above components, presented also in Figure 4-6. It needs to be noted that the interactions of the Knowledge Graph Manager with the Data Access Manager, and the Identity & Authorisation Manager, are not currently depicted.

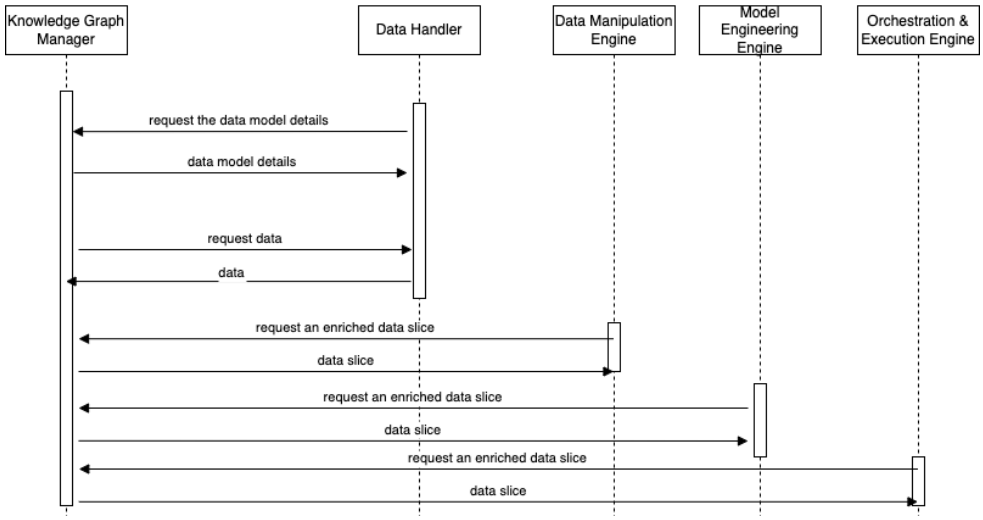

Figure 4-6: Knowledge Graph Manager – Interactions with other XMANAI Components

4.7 Model Engineering Engine

The Model Engineering Engine lies at the core of the AI Model Lifecycle Management Services Bundle and the AI Insights Services Bundle as it provides all functionalities related to the configuration, training and application of selected machine learning and deep learning models, as well as the explainability offerings of XMANAI. It is leveraged by data scientists and data engineers for the development of the respective models and processes, but also by business users as it plays an instrumental role in the provision of explanations for the model decisions. It is available and served through the Core AI Platform in the XMANAI Centralized Cloud deployment.

4.7.1 Overview

The Model Engineering Engine is the component responsible for managing all the operations regarding artificial intelligence models and their associated explainable tools. It offers a wide range of functionalities, addressing model development, evaluation and comprehension needs both during the experimentation and the production serving phase. The provided functionalities can be summarised as follows:

- Provision of machine learning and deep learning (including Graph ML) models that can be leveraged to address different analytics needs in the manufacturing domain. In close collaboration with the XAI Models Catalogue, the Model Engineering Engine allows the XMANAI users to create (train), retrieve and update (retrain) AI models. Additional functionalities are also foreseen to allow externally trained models to be leveraged within XMANAI.
- Configuration and execution of AI models' training sessions, including the definition of hyperparameters and selection of validation strategies. The datasets that will be used to train and evaluate the models, as well as the metrics that should be used for the model evaluation are defined through the component.
- Recording of AI experiments and logging of the relevant metadata. The term "experiment" in this context refers to any ML/DL model training process with a specific configuration of training data, metrics and hyperparameters. The Model Engineering Engine will allow comparison of experiments on the basis of the recorded parameters, metrics and artifacts and will further allow data scientists to retrieve the best hyperparameters.
- Safeguarding of the AI models' integrity and security, by employing risk assessment and mitigation techniques against adversarial attacks. Depending on the nature of the underlying model and the manufacturing problem it addresses, the component will enable detection of data poisoning attempts, adversarial training towards increased robustness, model corruption identification through checksums, etc.
- Provision of AI model explainability tools, libraries, and models to enable the generation of model decision explanations that address the diverse needs of the XMANAI stakeholders throughout the lifecycle of the corresponding AI pipelines. The Model Engineering Engine will thus provide explainability and interpretability insights to the data scientists during the model training and evaluation steps but will also offer targeted explanatory information to the business users, facilitating understanding of the underlying model decision mechanisms and impactful factors and thus contributing towards better operational decisions.

The operations performed by the Model Engineering Engine can be leveraged both independently, but also through their inclusion in more complex XMANAI AI pipelines, thus allowing their seamless combination with other data processing steps, indicatively provided by the Data Manipulation Engine. The mechanism that enables the extended pipeline creation and management is provided by the XAI Pipeline Manager (as described in Section 4.11).

The Model Engineering Engine consists of (a) the XAI Model Engineering Engine, (b) the Interactive Data Exploration and Experimentation Tool (partially), (c) the Experiment Tracking Engine, (d) the XAI Model Guard and (e) the XAI Model Explanations Engine that are described in detail in the XMANAI Deliverable D3.1.

4.7.2 Mapping to MVP Features & Technical Requirements

The Model Engineering Engine is designed taking into consideration different technical requirements defined in the XMANAI Deliverable D1.2. As depicted in the following table, the relevance of the related requirements is marked as high (if the Model Engineering Engine is mainly responsible for the specific functionality), medium (if the Model Engineering Engine contributes to the specific functionality) and low (if the Model Engineering Engine mostly leverages the specific functionality from other components in the XMANAI architecture).

Table 4-13: Model Engineering Engine - Mapping to Technical Requirements

No	As a…	I want to …	So that…	Relevance
TR_34	Business User, Data Scientist, Data Engineer	search and explore other data/AI assets on an in a user-friendly way (e.g. based on metadata, with sorting, filtering, matching level)	I can easily find what I am looking for	Low
TR_35	Data Scientist, Data Engineer	view metadata of the selected asset (e.g. datasets, AI models, AI pipelines, analysis results)	I can determine if the asset addresses my needs	Low
TR_44	Data Scientist, Data Engineer	have a control version of assets (AI models, AI pipelines, features, experiments, results)	I can keep track of the changes introduced and limit the impact of changes on existing pipelines	Low
TR_52	Data Scientist	preview a sample of the data	I can obtain a more concrete understanding of the data at hand	Low
TR_53	Data Scientist	view data distribution/profiling charts or summary statistics for the data (e.g. number of missing values, min/max values)	I can monitor data drifting issues	Low
TR_55	Data Scientist	create new features based on the current data (like min, max, mean values) that will be part of the same dataset	I can have more informative datasets, depending on the task	Medium
TR_57	Data Scientist	encode categorical data	I can prepare the data for the subsequent AI analysis	Medium
TR_58	Data Scientist	apply scaling and data normalization	I can prepare the data for the subsequent AI analysis	Medium
TR_59	Data Scientist	easily split the data for training and evaluation (data segmentation)	I can train and apply the models as I see fit	High
TR_60	Data Scientist	apply simple transformations on the data	I make them more appropriate for processing and visualisation	Low
TR_61	Data Scientist	change the data type of some features	I can manipulate them according to the needs of an AI model (e.g. convert to datetime)	Low

No	As a…	I want to ...	So that…	Relevance
TR_62	Data Scientist	apply data augmentation techniques	I can reduce overfitting of the models	Medium
TR_65	Data Scientist, Data Engineer	define and configure an AI pipeline for training, testing and/or production purposes	I can provide a solution for a specific problem	Medium
TR_66	Data Scientist, Data Engineer	include compatible baseline algorithms in an AI pipeline	I can provide a solution for a specific problem	Medium
TR_67	Data Scientist, Data Engineer	include compatible trained models in an AI pipeline	I can provide a solution for a specific problem	Medium
TR_68	Data Scientist	register a trained AI model I have created as part of an AI pipeline	I can reuse it in my AI pipelines	Medium
TR_75	Data Scientist	configure training to control parameters, such as learning rate reduction when a metric has stopped improving or stop it	I can get notified if there is any problem during training	High
TR_76	Data Scientist	receive recommendations for automated feature selection	I can perform feature engineering in a faster / easier manner, in cases of high-dimensionality data	High
TR_77	Data Scientist	choose among different methods to apply for explaining an AI pipeline (including its input data, models and results)	I can select the ones that best fit with the problem I am solving	High
TR_78	Data Scientist	properly adjust the explanations depending on the user profile	I can increase understanding of the results for its intended users	High
TR_84	Business User	properly visualise the explanations depending on the user profile	I can adapt the information I receive, according to my needs	Medium
TR_85	Data Scientist	define the parameters and metrics of the experiments associated with an AI pipeline	I can track and compare my experiments	High
TR_86	Data Scientist	run automatic tests on the registered AI models (when including them in an AI pipeline)	I can quickly check that the trained models are robust and fault-tolerant	High
TR_87	Data Scientist	support the inclusion of different performance metrics as part of an AI pipeline	I can obtain a better picture of my pipelines' / models' effectiveness according to my needs	High
TR_88	Data Scientist	generate multiple visualisations as output of an AI pipeline	I can make it available to the involved users	Medium
TR_89	Business User	view a visualisation including the results and their explanations	I can take informed decisions	Low
TR_91	Business User	request explanations for why certain predictions were generated in an AI pipeline	I can trust the model	Medium
TR_92	Data Scientist	respond to requests for explanations of an AI pipeline generated by users	I can help the business users to trust the model	High

No	As a…	I want to ...	So that…	Relevance
TR_93	Data Scientist	have a common metadata model for describing my AI pipelines, experiments, results	I can share them with other users	Medium
TR_94	Business User	define model evaluation metrics (beyond the ones used by the data scientists)	I can monitor how the model's performance translates to my business KPIs	Medium
TR_95	Business User	retrieve and compare previous AI model's results with the real outcomes once they are available	I can keep track of the model's performance over time	High
TR_97	Data Scientist	track the performance of my trainings (regarding stability and converge of the results, execution times, the use of computational resources, etc.)	I can be automatically informed of non-normal scenarios via descriptive error messages	Medium
TR_98	Data Scientist	implement different mechanisms to save a checkpoint of the AI model	I can resume the training of a model from the previous point	High
TR_99	Data Scientist	know when the hyperparameters of the model are optimized	I can avoid overfitting	High
TR_101	Data Scientist, Data Engineer	get descriptive error messages	I can debug the AI pipelines that have been executed	Medium
TR_102	Data Scientist, Data Engineer	view detailed execution logs for an AI pipeline	I can detect bottlenecks once my AI pipeline has been executed	Medium
TR_103	Data Scientist, Data Engineer	keep a history of experiments performed (pipeline runs)	I can compare results & improve the pipeline	High
TR_109	Data Scientist	create reports of the performed data analysis locally on my environment	I can evaluate the results	Medium
TR_114	Data Scientist	provide wrappers (register) for baseline algorithms from selected ML libraries (e.g. sk-learn, spark mllib)	I can include them in my AI pipeline	Medium
TR_115	Data Scientist	provide wrappers (register) for baseline algorithms for clustering, classification, regression, dimensionality reduction	I can include them in my AI pipeline	Medium
TR_116	Data Scientist	provide wrappers (register) for baseline algorithms from selected DL libraries (e.g. tensorflow)	I can include them in my AI pipeline	Medium
TR_117	Data Scientist	provide wrappers (register) for explainability techniques	I can include them in my AI pipeline	Medium
TR_118	Data Scientist	have a common metadata model for describing my trained models (hyperparameters, parameters, code, metrics...)	I can share them with other users	Medium
TR_119	Data Scientist	package the trained AI models I want to register (following specific guidelines for directory tree, programming language, name of files, etc.)	I can upload them to the XMANAI catalogue	Medium
TR_120	Data Scientist	register a trained AI model I have created offline to solve a specific problem	I can make it available to be reused by other users	Medium

No	As a…	I want to …	So that…	Relevance
TR_121	Data Scientist	keep versioning for the trained AI model I have created offline	I can retrain the model without affecting all AI pipelines it has been reused	Low
TR_122	Data scientist	follow specific guidelines for the explainability techniques (e.g. surrogate models) I want to register	all techniques and their associated metadata (including python packages requirements) can be packaged and available in the AI pipelines	Medium
TR_123	Data Scientist	register a surrogate model I have created offline	I can reuse it in my AI pipelines	Medium
TR_124	Data Scientist	know which validation method is right for my AI model	I can validate the model(s) with higher accuracy	High
TR_125	Data Scientist	use various performance metrics for the AI models	I have a better picture of my models' effectiveness	High
TR_126	Data Scientist	generate an evaluation report for each trained ML/DL model	users that insert it in their AI pipelines are aware of its performance	High
TR_127	Business User (Expert)	evaluate the validity of the explanations and provide feedback	I can improve the AI models and AI pipelines to solve a specific problem	Medium
TR_128	Data Scientist	compare results from different models created for a particular task	I can gain an understanding of which factors aid in the performance of the models (which features helped, how the different preprocessing steps affected the result	High
TR_129	Data Scientist	compare performances of different AI pipelines	I can track the improvements of the different versions of my pipelines / models	Medium
TR_130	Business User	view reports of the experiments (simulations of different settings, models, and methods) to solve a specific problem	I can evaluate the results	High
TR_131	Data Scientist	test the training data sets at each step to detect possible poisoned data points	I can check the integrity of data sets	High
TR_132	Data Scientist	filter poisoned data points out and retrain the model	I can repair possible poisoning of the model	Medium
TR_133	Data Scientist, Data Engineer	generate adversarial examples	I can create a robust model against adversarial attacks	High
TR_134	Data Scientist	check training data sets for possible unfair biases	I can prevent possible discriminatory biases of the model	Medium

In addition, the MVP features that are within the scope of the Model Engineering Engine are depicted in the following table. It needs to be noted that the bold highlights indicate the MVP features for which the Model Engineering Engine is primarily responsible.

Table 4-14: Model Engineering Engine - Mapping to MVP features

ID	Title
XMANAI_F_DM_017	**AI Model Management (create/store/update/delete/clone/export/import, configure, versioning, register/import)**
XMANAI_F_DM_018	AI Pipeline Management (create/store/update/delete/clone/export/join, view IPR of assets involved, define templates, versioning)
XMANAI_F_DM_023	**AI Model Security Assessment**
XMANAI_F_DP_032	Data Transformation (normalisation, encoding, modifying data types, etc.)
XMANAI_F_AI_033	**AI Model Design (define, configure, store, import, export)**
XMANAI_F_AI_034	**AI Model Training, Application & Evaluation (experimentation vs production, configure control parameters, define/monitor eval . metrics, save check points, support parameter optimisation)**
XMANAI_F_AI_035	AI Pipeline Design (define, configure, register AI model, add annotations/comments, reuse common features)
XMANAI_F_AI_036	AI Pipeline Execution & Evaluation on XMANAI Common Cloud (experimentation vs production, run automatic tests for AI models, run scheduling, configuration)
XMANAI_F_AI_037	AI Pipeline Execution & Evaluation on Premise / Private Cloud (experimentation vs production, run automatic tests for AI models, run scheduling, configuration)
XMANAI_F_EX_042	**Explainability Methods Management (add/remove/configure, register/import)**
XMANAI_F_EX_043	**Collaboration over AI model/results/pipelines explanations (application of explainability methods at AI pipeline or model level, results querying)**
XMANAI_F_EX_044	Explainability Results Visualisation (various charts, adjust based on user profile)
XMANAI_F_EXI_045	Explainability Results Evaluation (allow manual feedback & results validation)

4.7.3 Interactions

In terms of expected interactions with other XMANAI components, the Model Engineering Engine has (or leverages) interfaces with the following components:

- The Data Access Manager indirectly for ensuring that access to data, results and model training artifacts, including the trained models, is performed according to the user defined policies.
- The Knowledge Graph Manager through which it retrieves the data that will be used to train and/or to apply the AI models. The way result data can be stored after the execution of the model engineering steps also requires interactions between the two components.
- The XAI Models Catalogue which provides the XMANAI AI models and therefore interacts with the current component for all model handling functionalities, including retrieving trained models and updating models that have been retrained or their configuration has been in any way altered.
- The Execution & Orchestration Engine both directly or indirectly depending on whether the component's functionalities are integrated in an AI pipeline or the direct execution invocation functionalities are used instead.
- The Identity & Authorisation Manager, similar to the Access Manager, is indirectly leveraged to ensure all users can smoothly perform the operations foreseen for them and for their organisations.
- The Data Manipulation Engine may provide direct input to the Model Engineering Engine in cases when the pre-processed data are fed to a model-related function, e.g., model training.
- The Provenance Engine for tracking the lineage of the data processed by the Model Engineering Engine.

- The XAI Pipeline Manager for handling the inclusion of model engineering tasks in the overall AI pipelines, a process that includes numerous interactions, indicatively including the retrieval of model training tasks' configuration, the invocation of the execution services for the respective tasks, the retrieval and exchange of the corresponding logs etc.
- The XAI Visualisation Engine, either directly or indirectly, again depending on the mode that is utilized, i.e., through an AI pipeline or independently. Many of the execution results of the configured and executed model engineering functionalities are best consumed in a visual way, thus interaction with the Visualization Engine to configure the appropriate charts is foreseen.

The collaboration of the current component with the rest XAI components is also presented in the following diagram. It needs to be noted that the interactions of the Model Engineering Engine with the Data Access Manager, and the Identity & Authorisation Manager, are not currently depicted in Figure 4-7.

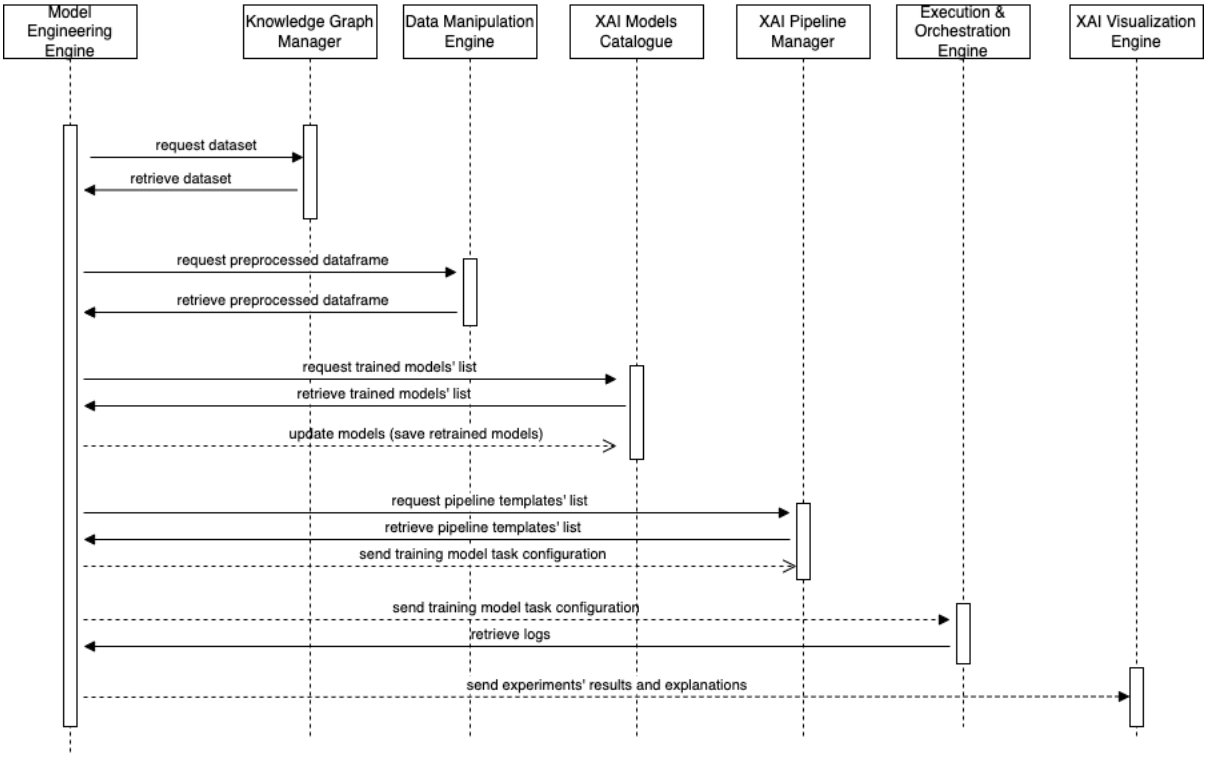

Figure 4-7: Model Engineering Engine – Interactions with other XMANAI Components

4.8 Provenance Engine

As indicated by its name, the Provenance Engine is responsible for managing the provenance-related information accompanying a data asset, and is associated with the Data & Models Collection Services Bundle in XMANAI. The Provenance Engine runs under the hood in the Centralized Cloud and the On-Premise (Private Cloud) instances of the XMANAI Platform.

4.8.1 Overview

A Provenance Engine has the central task of tracking and making changes to the datasets traceable. For this purpose, different versions of the relevant datasets must be created. Tracking changes only saves the modifications line by line in a text-based file. Binary files cannot be versioned in this way. For them, a completely new version must always be saved.

For traceable change tracking, the datasets and their modifications should be saved and the transactions that resulted in a dataset revision. This means that - in addition to saving datasets version

- scripts, programmes, and other commands must also be saved so that a user can understand how the data has been transformed. These transactions must then also be versioned.

The second part of a provenance engine - besides versioning - is about storage of metadata for each version of a dataset. A metadata manager is needed that tracks information about the different versions of data as well as the transactions performed on the datasets. For this purpose, the World Wide Web Consortium has published a W3C Recommendation that uses a generic approach to specify and describe a data model. With this ontology, it is possible to store the provenance of data in a Triplestore or another graph database.

Such a component is essential for a data science platform because users need to transform data accordingly for certain models and specific purposes. Often, data is not available in the desired granularity or is additionally aggregated. This leads to massive changes in the data sets in some cases. If errors occur later in the analyses, they can be identified more efficiently with the help of a provenance engine.

An implementation should be thought of from the beginning. Once new data is added or manipulated, a data provenance engine should take over versioning and metadata management. For a user, it is helpful if these tasks are carried out in the background without his involvement. Nevertheless, interfaces should also exist for a user to control some processes of the provenance engine.

4.8.2 Mapping to MVP Features & Technical Requirements

Table 4-15 below shows the relevant technical requirements related to the Data Provenance Engine component. These technical requirements are from Deliverable 1.2 and have been extended by the column "Relevance". "High" means that the component is primarily responsible for or centrally involved in executing the tasks. "Medium" means that the component contributes to the specific functionality.

Table 4-15: Provenance Engine - Mapping to Technical Requirements

No	As a…	I want to …	So that…	Relevance
TR_42	Data Scientist, Data Engineer	view how often the data is updated and when	I can know if I have the latest data or change the update frequency if needed	High
TR_43	Data Scientist	have a control version of the datasets	I am sure that I use the latest updated data	High
TR_44	Data Scientist, Data Engineer	have a control version of assets (AI models, AI pipelines, features, experiments, results)	I can keep track of the changes introduced and limit the impact of changes on existing pipelines	High
TR_45	Business user	XMANAI to register each access event (based on actions performed) of other users to my data or other assets	I can have detailed logs who and when accessed my data and other assets	High
TR_46	Business user	see a list of users who has ever had access to an asset that I provided and what activities were performed	I can monitor my assets usage	High
TR_47	Business user	view which data and assets I have shared with whom	I can monitor my sharing activities	High
TR_48	Business user	XMANAI to register to whom I permitted access to my data or other assets	I can log the sharing my data and other assets with other users	High

TR_49	Data Scientist	check the IPR of the assets involved in an AI pipeline	all assets included in a pipeline are used in alignment with their licenses	Medium

In addition, the MVP features that are within the scope of the Provenance Engine are depicted in the following table.

Table 4-16: Provenance Engine - Mapping to MVP Features

ID	Title
XMANAI_F_DM_014	Dataset Management (add/edit/remove asset and metadata, versioning)
XMANAI_F_DM_015	Features Management (store, add/edit/remove asset and metadata, define rules, versioning)
XMANAI_F_DM_016	Results Management (edit/remove asset and metadata, versioning)
XMANAI_F_DM_017	AI Model Management (create/store/update/delete/clone/export/import, configure, versioning, register/import)
XMANAI_F_DM_018	AI Pipeline Management (create/store/update/delete/clone/export/join, view IPR of assets involved, define templates, versioning)

4.8.3 Interactions

Looking at the data handler, CRUD operations are discussed in more detail in this section. When a user adds new datasets, intermediate results are created, or models are developed, they are generally saved. Each time a dataset is updated or created, a new version is also created, and the update is saved. A record of the metadata is stored in the corresponding PROV Store.

In the case of a delete, the datasets are deleted completely. However, the metadata is preserved; likewise, the different dataset versions are preserved so that a reset can be realised.

If information is to be requested, the metadata can provide details about the changes. In addition, it is possible that an older version can also be given back so that analyses or errors can be examined, e.g. due to performance or bugs. Thus, a restore also falls under this function.

It needs to be noted that the interactions of the Provenance Engine with the Data Access Manager, and the Identity & Authorisation Manager, are not currently depicted in Figure 4-8.

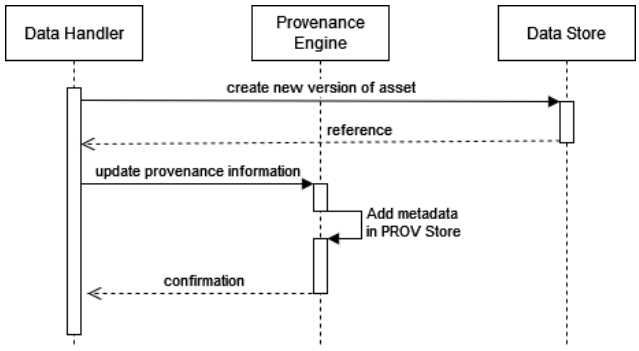

Figure 4-8: Provenance Engine – Interaction with Data Handler by creating / updating /deleting asset operation

4.9 XAI Marketplace

The XAI Marketplace belongs to the Secure Asset Sharing Services bundle, and it facilitates the exchange of assets between different users (business users, data scientists, etc) and organizations. It is available and served through the Core AI Platform in the XMANAI Centralized Cloud deployment.

4.9.1 Overview

The XAI Marketplace is responsible for providing a user with the means to view, search and discover data assets of interest and then, facilitating the process that involves the acquisition of one or more data assets based on a mutually agreed contract between the asset provider and the asset consumer. In general, data assets may vary in nature, and can belong to one of the following categories:

- Datasets (in raw or processed form)
- AI models (in a baseline or trained format)
- Experimental (explainable) results
- Full, ready-to-be-served, XAI pipelines

Each data asset is described by a set of metadata adapted to its type, in order to make it easier (both for the users and for the XMANAI components, under the hood) to find, use and manage it. The XAI Marketplace handles, stores and maintains the metadata and utilizes them when a user browses the marketplace catalogue or is in search for assets of interest. The content and features that the marketplace interface provides to the user is, of course, in alignment with each asset's policies and IPRs as provided by the Data Access Manager. For example, a user of an organization can view the detailed profile of a data asset that belongs to the same organization, and immediately access it. On the other hand, a user from a different organization that encounters the same asset will view only basic metadata details and, if allowed, can make a request to acquire it. It should be noted that the metadata information is also accessible and retrievable from other XMANAI components that manage assets.

As a place of interaction between users, the XAI Marketplace, relies heavily on a protected environment with a clearly defined contract management and monitoring system, that provides the appropriate conditions for reliable data asset sharing between different organizations, or between departments and users of the same organisation. When a potential consumer expresses interest on an asset, the XAI Marketplace allows the asset owner to draft, edit and manage an agreement in a machine-readable format. Through the XAI Marketplace interface the involved parties can negotiate over the terms and conditions, and monitor the status of the whole process, which will hopefully reach a point that satisfies all the involved parties. The XAI Marketplace is then responsible to enforce the agreed terms, which are safely stored in the XMANAI distributed ledger in the form of a smart contract.

The XAI Marketplace consists of the following components: (a) the Registry/Metadata Manager & Metadata Store, and (b) the Contract Manager. These components are adequately described in the XMANAI Deliverable D2.1.

4.9.2 Mapping to MVP Features & Technical Requirements

The technical requirements related to the XAI Marketplace (as initially defined in the XMANAI Deliverable D1.2) are presented in the table below. Each requirement is characterized by its degree of relevance to this component. High relevance means that the XAI Marketplace is mainly responsible for the specific functionality, medium relevance implies that the XAI Marketplace contributes to the specific functionality and low relevance is when the XAI Marketplace mostly leverages the specific functionality from other components in the XMANAI architecture.

Table 4-17: XAI Marketplace - Mapping to Technical Requirements

No	As a…	I want to …	So that…	Relevance
TR_12	Data Scientist, Data Engineer, Business User	add/edit metadata for my data (based on a common metadata model)	I can improve the quality of data for further reuse	High

No	As a…	I want to …	So that…	Relevance
TR_28	Data Scientist, Data Engineer, Business User	ensure that only properly authenticated users have access to my assets (datasets, AI models, AI pipelines, experiments, analysis results)	I can ensure their privacy and security	Low
TR_31	Business user, Data Scientist	ensure that my data are transferred between the different layers of the platform securely	I can ensure that my data will not be disclosed to unauthorized parties	Medium
TR_32	Business User, Data Scientist, Data Engineer	share my assets (e.g. datasets, AI models, features, AI pipelines, experiments, analysis results) with other organisations / users of my preference	I can help someone else who possibly needs this information	High
TR_33	Business User	trade my assets (e.g. datasets, AI models, features, AI pipelines, experiments, analysis results) in a secure and reliable way for a specific time period	I can gain new revenues from my assets	High – not prioritized in the MVP though
TR_34	Business User, Data Scientist, Data Engineer	search and explore other data/AI assets on an in a user-friendly way (e.g. based on metadata, with sorting, filtering, matching level)	I can easily find what I am looking for	High
TR_35	Data Scientist, Data Engineer	view metadata of the selected asset (e.g. datasets, AI models, AI pipelines, analysis results)	I can determine if the asset addresses my needs	High
TR_36	Business User, Data Scientist, Data Engineer	know which are the IPR holders involved in the asset I am interested in and what are their associated rights/licenses	I can take an informed decision for the asset acquisition	High
TR_37	Business User, Data Scientist, Data Engineer	get access to assets (e.g. datasets, AI models, features, AI pipelines, experiments, analysis results) created by other users, in a secure and reliable way for a specific time period	I can enrich my data and enhance the analytics results for my company	High
TR_38	Business User, Data Scientist, Data Engineer	buy assets (e.g. datasets, AI models, features, AI pipelines, experiments, analysis results) created by other users, in a secure and reliable way for a specific time period	I can enrich my data and enhance the analytics results for my company	High – not prioritized in the MVP though
TR_39	Business User, Data Scientist, Data Engineer	manage the contracts for sharing/trading my assets and configure terms	I can have full control on the potential agreement	High
TR_40	Business user, Data Scientist, Data Engineer (and asset provider)	be able to negotiate a sharing/trading agreement in a flexible and reliable manner	I can achieve the best outcome for my company	High
TR_49	Data Scientist	check the IPR of the assets involved in an AI pipeline	all assets included in a pipeline are used in alignment with their licenses	Low
TR_93	Data Scientist	have a common metadata model for describing my AI pipelines, experiments, results	I can share them with other users	Medium

No	As a…	I want to …	So that…	Relevance
TR_118	Data Scientist	have a common metadata model for describing my trained models (hyperparameters, parameters, code, metrics...)	I can share them with other users	Medium

Table 4-18 presents the MVP features, also initially presented in D2.1, that are relevant to the scope of the XAI Marketplace. The MVP features for which the XAI Marketplace is primarily responsible are depicted in bold fonts.

Table 4-18: XAI Marketplace - Mapping to MVP Features

ID	Title
XMANAI_F_DM_014	Dataset Management (add/edit/remove asset and metadata, versioning)
XMANAI_F_DM_016	Results Management (edit/remove asset and metadata, versioning)
XMANAI_F_DM_017	AI Model Management (create/store/update/delete/clone/export/import, configure, versioning, register/import)
XMANAI_F_DM_018	AI Pipeline Management (create/store/update/delete/clone/export/join, view IPR of assets involved, define templates, versioning)
XMANAI_F_DM_020	Data Asset Access Policies and Licencing (configuration, management and enforcing)
XMANAI_F_DM_021	Data Asset Secure Transfer (through platform layers, operation and enforcing)
XMANAI_F_DS_025	**Data Asset Sharing (based on contracts, with selected users / organizations)**
XMANAI_F_DS_026	**Data Asset Trading (based on contracts, payment performed offline/ online)**
XMANAI_F_DS_027	**Data Asset Contract Management (contract preparation, negotiation, agreement, enforcement)**
XMANAI_F_DS_028	**Data Asset Search and Discovery (including indexing, sorting, filtering, matching level to your asset, view details)**
XMANAI_F_AI_039	AI Pipeline Results Management (store, export, add summary statistics, easily compare with real values (for predictions), retrieve via API)

4.9.3 Interactions

The XAI Marketplace is about sharing data assets, and as such, it is expected to interact with other XMANAI components that manage or produce sharable assets or handle their properties. In general, the XAI Marketplace has (or leverages) interfaces with the following components:

- The Data Access Manager for retrieving the access policies for each XAI Asset.
- The Provenance Engine for tracking the lineage of the data and models displayed or searched via the XAI Marketplace interface.
- The Data Handler for receiving the initial information for a dataset that has been ingested into the platform.
- The XAI Models Catalogue for retrieving the different ML/DL models, to be shared with other users.
- The XAI Pipeline Manager for retrieving fully configured and successfully executed XAI pipelines, also to be shared with other users.

Of course, there are also interactions with the Identity Manager for user authorization and authentication, but these are very typical interactions, commonly encountered in software solutions. Thus, they are omitted from the following figure which presents the XAI Marketplace interactions with other components in more detail.

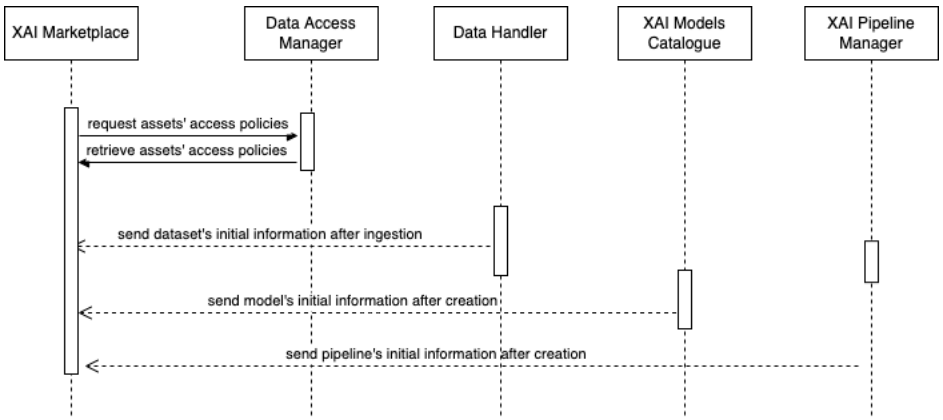

Figure 4-9: XAI Marketplace – Interactions with other XMANAI Components

4.10XAI Models Catalogue

The XAI Model Catalogue is a component of the XMANAI Platform that belongs to the AI Model Lifecycle Management Services Bundle. It is responsible for the different management tasks related to the XAI models offered by XMANAI.

4.10.1Overview

The XAI Model Catalogue provides the services to add, update and delete models (and their associated metadata), store different experiments and list all the different models, experiments and related metadata. As part of the AI Lifecycle Management Service, it provides a set of different functionalities:

- Add a new model to the platform. A model is composed of a code part and a metadata part. A Python class of the model and its requirements file constitute the code part. The model class contains a set of methods covering at least the following tasks: hyperparameters assignment (covered by the __init__ function), a method responsible for checking the format of the input data and performing all necessary processing in case it needs to be adapted to the input format of the ML/DL model, a function used to perform the model training, a method responsible for validating the performance of the trained model and an inference function used to generate inferences from a trained model, given a specific input and a specific version of the model. The requirements file contains a list of Python packages required to be able to use the model. The associated metadata includes general information about the model, considering at least: a short description explaining the model to the end user, a flag to mark a model as a Graph Model, the type of learning (whether it is supervised, unsupervised, etc.), the category of the model (classification, regression model, etc.), a flag to consider a model retrainable from a previous version, the type and format of the input data, the type and format of the predictions, the configurable hyperparameters of the model, and the accepted validation metrics that will be also available through the XAI Marketplace. The access policies of the model also need to be defined and enforced in collaboration with the Data Access Manager.
- Add a new explainability technique. Similar to the uploading of new models, this functionality allows the user to submit a new explainability method, adding the code needed to be executed, the requirements of the python packages required to be run and the associated metadata: selection of post-hoc or surrogate model, models that are compatible, and configurable hyperparameters in case of a surrogate model.
- Update a model/explainability technique. This functionality refers to the update of one of the component parts of a model or explainability method, either the code or the metadata. The updated model will be stored as a new version of the same model or explainability tool.

- Delete a model/explainability technique. Through this functionality, a specific version of a model or explainability method can be removed from the platform, both the code and the associated metadata.
- List of models, experiments and explainability techniques. This functionality allows to send a list of all components belonging to one of these three groups to other components of the XMANAI architecture.

4.10.2 Mapping to MVP Features & Technical Requirements

The XAI Models Catalogue is designed taking into consideration different technical requirements defined in the XMANAI Deliverable D1.2. As depicted in the following table, the relevance of the related requirements is marked as high (if the XAI Models Catalogue is mainly responsible for the specific functionality), medium (if the XAI Models Catalogue contributes to the specific functionality) and low (if the XAI Models Catalogue mostly leverages the specific functionality from other components in the XMANAI architecture).

Table 4-19: XAI Models Catalogue - Mapping to Technical Requirements

No	As a…	I want to …	So that…	Relevance
TR_44	Data Scientist, Data Engineer	have a control version of assets (AI models, AI pipelines, features, experiments, results)	I can keep track of the changes introduced and limit the impact of changes on existing pipelines	High
TR_66	Data Scientist, Data Engineer	include compatible baseline algorithms in an AI pipeline	I can provide a solution for a specific problem	Medium
TR_67	Data Scientist, Data Engineer	include compatible trained models in an AI pipeline	I can provide a solution for a specific problem	Medium
TR_75	Data Scientist	configure training to control parameters, such as learning rate reduction when a metric has stopped improving or stop it	I can notified if there is any problem during training	Low
TR_86	Data Scientist	run automatic tests on the registered AI models (when including them in an AI pipeline)	I can quickly check that the trained models are robust and fault-tolerant	Low
TR_114	Data Scientist	provide wrappers (register) for baseline algorithms from selected ML libraries (e.g. sk-learn, spark mllib)	I can include them in my AI pipeline	High
TR_115	Data Scientist	provide wrappers (register) for baseline algorithms for clustering, classification, regression, dimensionality reduction	I can include them in my AI pipeline	High
TR_116	Data Scientist	provide wrappers (register) for baseline algorithms from selected DL libraries (e.g. tensorflow)	I can include them in my AI pipeline	High
TR_117	Data Scientist	provide wrappers (register) for explainability techniques	I can include them in my AI pipeline	High
TR_118	Data Scientist	have a common metadata model for describing my trained models (hyperparameters, parameters, code, metrics...)	I can share them with other users	Medium
TR_119	Data Scientist	package the trained AI models I want to register (following specific guidelines for directory tree, programming language, name of files, etc.)	I can upload them to the XMANAI catalogue	High

No	As a…	I want to …	So that…	Relevance
TR_120	Data Scientist	register a trained AI model I have created offline to solve a specific problem	I can make it available to be reused by other users	High
TR_121	Data Scientist	keep versioning for the trained AI model I have created offline	I can retrain the model without affecting all AI pipelines it has been reused	Medium
TR_122	Data scientist	follow specific guidelines for the explainability techniques (e.g. surrogate models) I want to register	all techniques and their associated metadata (including python packages requirements) can be packaged and available in the AI pipelines	High
TR_123	Data Scientist	register a surrogate model I have created offline	I can reuse it in my AI pipelines	High
TR_125	Data Scientist	use various performance metrics for the AI models	I have a better picture of my models' effectiveness	Medium

In addition, the MVP features that are within the scope of the XAI Models Catalogue are depicted in the following table. It needs to be noted that the bold highlights indicate the MVP features for which the XAI Models Catalogue is primarily responsible.

Table 4-20: XAI Models Catalogue - Mapping to MVP Features

ID	Title
XMANAI_F_DM_015	Features Management (store, add/edit/remove asset and metadata, define rules, versioning)
XMANAI_F_DM_016	Results Management (edit/remove asset and metadata, versioning)
XMANAI_F_DM_017	**AI Model Management (create/store/update/delete/clone/export/import, configure, versioning, register/import)**
XMANAI_F_DM_018	AI Pipeline Management (create/store/update/delete/clone/export/join, view IPR of assets involved, define templates, versioning)
XMANAI_F_AI_034	AI Model Training, Application & Evaluation (experimentation vs production, configure control parameters, define/monitor eval. metrics, save check points, support parameter optimisation)
XMANAI_F_AI_035	AI Pipeline Design (define, configure, register AI model, add annotations/comments, reuse common features)
XMANAI_F_AI_036	AI Pipeline Execution & Evaluation on XMANAI Common Cloud (experimentation vs production, run automatic tests for AI models, run scheduling, configuration)
XMANAI_F_AI_037	AI Pipeline Execution & Evaluation on Premise / Private Cloud (experimentation vs production, run automatic tests for AI models, run scheduling, configuration)
XMANAI_F_EX_042	**Explainability Methods Management (add/remove/configure, register/import)**

4.10.3 Interactions

In terms of expected interactions with other XMANAI components, the XAI Models Catalogue has (or leverages) interfaces with the following components:

- The Model Engineering Engine, which has to interact with the catalogue to retrieve the list of trained models (and their associated metadata), explainability tools or to add a new experiment (trained model and metadata resulting from training) in the catalogue.
- The XAI Pipeline Manager, where a list of all models stored in the catalogue will be sent when required.

- The XAI Marketplace where the catalogue of all data assets (including XAI models) is presented.

The collaboration of the current component with the rest XAI components is also presented in the following diagram. It needs to be noted that the XAI Models Catalogue also interacts with the Data Access Manager, and the Identity & Authorisation Manager, even though the specific interactions are not currently depicted in Figure 4-10.

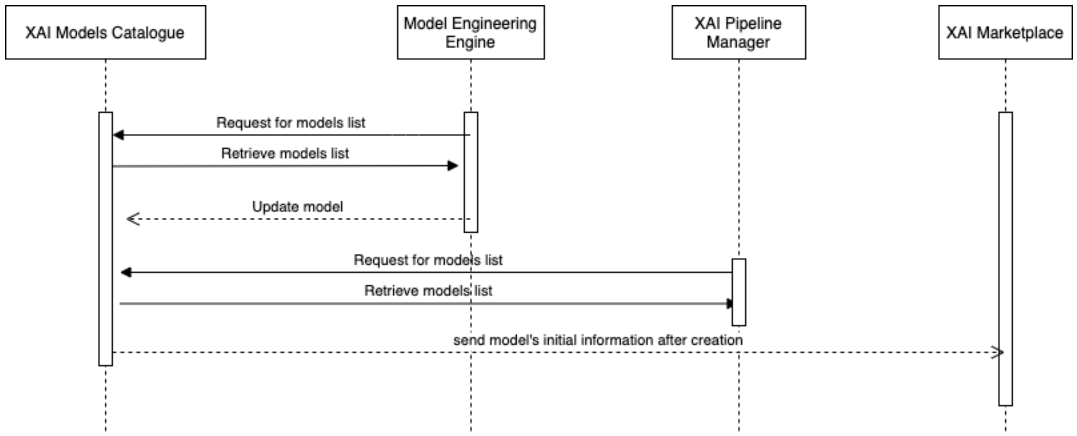

Figure 4-10: XAI Models Catalogue – Interactions with other XMANAI Components

4.11XAI Pipeline Manager

The XAI Pipeline Manager has an instrumental role within the AI Model Lifecycle Services Bundle and the AI Insights Services Bundle as it enables the design of Explainable AI Pipelines by data scientists and data engineers. It is available and served through the Core AI Platform in the XMANAI Centralized Cloud deployment.

4.11.1Overview

The XAI Pipeline Manager is responsible for the design, management, execution and monitoring of **Explainable AI Pipelines** that orchestrate in an end-to-end manner the flow of data into, and output from, a machine learning or deep learning model (or set of multiple models). Explainable AI Pipelines essentially refer to robust, independently executed workflows of different types of data-related tasks, that have been appropriately arranged in a directed acyclic graph (DAG), including:

- Data manipulation tasks, that indicatively refer to cleaning, transformation, manipulation, and normalization of the raw data input in order to perform the necessary feature engineering tasks and ensure that the data are appropriate for running an analysis. Such data processing tasks are handled in collaboration with the Data Manipulation Engine.
- AI model tasks that embrace the training, validation, application and evaluation tasks of machine learning or deep learning algorithms. Such model-related tasks are created in collaboration with the Model Engineering Engine, leveraging the XAI Models Catalogue.
- Input tasks, that handle the retrieval of the appropriate data slice that is required (in collaboration with the Knowledge Graph).
- Output tasks that concern how the results of the pipeline execution are stored and retrieved (in order to be subsequently used by the XAI Visualization Engine).

Explainable AI Pipelines bring different “explainability” aspects at data, model and result levels in the foreground in order to ensure a common and shared understanding among all involved stakeholders, thus each task needs to implicitly or explicitly address how explainability is tackled, as a pre-condition or post-condition of its execution. For example, any data manipulation task has data explainability as

a pre-condition since the data structure and semantics needs to be known in advance for the raw data input; any AI model task has model and/or result explainability as a post-condition since different explainability techniques shall be leveraged in a model-agnostic manner, depending on the algorithm that has been selected; for output tasks, result explainability is of utmost importance to ensure that the XAI pipelines predictions are accompanied by appropriate explanations to shed light on different questions the target audience of the predictions may have. In alignment with the Model Engineering Engine, the XAI Pipeline manager supports a number of open-source machine learning libraries, deep learning libraries and explainability techniques.

In principle, each pipeline belongs to a single user and shared within an organization, yet it is expected that different users may contribute by effectively collaborating on the configuration of the pipeline's tasks, and by providing annotations and comments to clarify its overall use or the explanations provided. At any moment, an XAI Pipeline has a state that may range from draft (under configuration) to ongoing (being discussed with different stakeholders or tested with sample data), finalized (ready for execution) and deployed (to be used in production).

The XAI Pipeline Manager allows users to create custom pipelines using a drag-and-drop interface (without requiring or permitting any coding) and indicating the dependencies between different tasks, which results in a dynamic execution graph. Any XAI pipeline that has been created in the XAI Pipeline Manager can be managed and scheduled for execution (by the Execution & Orchestration Engine), taking into consideration that the pipeline's settings extend over: (a) the execution location, namely in the centralized cloud (provided by the XMANAI Platform) or in a private cloud instance (that is already available in certain XMANAI demonstrators); (b) the execution modality that concerns the schedule when the execution should be automatically triggered or the circumstances under which execution is triggered on-demand (e.g. by the XMANAI manufacturing apps). Through the XAI Pipeline Manager user interface, the users are also able to monitor the execution of their XAI pipelines and track the metrics they are interested to follow.

The XAI Pipeline Manager consists of: (a) the Pipeline Designer, and (b) the Pipeline Serving & Monitoring Engine that are described in detail in the XMANAI Deliverable D3.1

4.11.2 Mapping to MVP Features & Technical Requirements

The XAI Pipeline Manager is designed taking into consideration different technical requirements defined in the XMANAI Deliverable D1.2. As depicted in the following table, the relevance of the related requirements is marked as high (if the XAI Pipeline Manager is mainly responsible for the specific functionality), medium (if the XAI Pipeline Manager contributes to the specific functionality) and low (if the XAI Pipeline Manager mostly leverages the specific functionality from other components in the XMANAI architecture).

Table 4-21: XAI Pipeline Manager - Mapping to Technical Requirements

No	As a…	I want to ...	So that…	Relevance
TR_65	Data Scientist, Data Engineer	define and configure an AI pipeline for training, testing and/or production purposes	I can provide a solution for a specific problem	High
TR_66	Data Scientist, Data Engineer	include compatible baseline algorithms in an AI pipeline	I can provide a solution for a specific problem	High
TR_67	Data Scientist, Data Engineer	include compatible trained models in an AI pipeline	I can provide a solution for a specific problem	High
TR_68	Data Scientist	register a trained AI model I have created as part of an AI pipeline	I can reuse it in my AI pipelines	Medium
TR_69	Data Scientist, Data Engineer	collaborate in the configuration of AI pipelines with selected users (within	I can create optimal workflows for a specific problem	High

No	As a…	I want to …	So that…	Relevance
		my organization or external to my organisation)		
TR_70	Data Scientist, Data Engineer	define pipelines that can be used as templates for specific problems	my colleagues and I can reuse them	High
TR_71	Data Scientist, Data Engineer	clone a designed pipeline	I can create alternative version of the pipeline and improve its performance without re-creating it from scratch	High
TR_72	Data Scientist, Data Engineer	join designed pipelines	I can create advanced combinations (e.g. training pipeline with testing pipeline, multiple training pipelines)	High
TR_73	Data Scientist, Data Engineer	execute step-by-step an AI pipeline over sample data	I can ensure that the result is the intended one	High
TR_74	Data Scientist, Data Engineer	reuse common features in different pipelines	I do not need to recompute them	Medium
TR_75	Data Scientist	configure training to control parameters, such as learning rate reduction when a metric has stopped improving or stop it	I can notified if there is any problem during training	Medium
TR_76	Data Scientist	receive recommendations for automated feature selection	I can perform feature engineering in a faster / easier manner, in cases of high-dimensionality data	Medium
TR_77	Data Scientist	choose among different methods to apply for explaining an AI pipeline (including its input data, models and results)	I can select the ones that best fit with the problem I am solving	Medium
TR_78	Data Scientist	properly adjust the explanations depending on the user profile	I can increase understanding of the results for its intended users	Medium
TR_79	Data Scientist	define summary statistics to be computed for an AI pipeline or part of it	I can explain the behavior of the inputs and / or outputs of a pipeline / model	High
TR_80	Data Scientist	add annotations and comments in AI pipelines	I can better explain the results	High
TR_81	Data Scientist	define whether the analysis results should be saved as a new dataset or update an existing one	I can re-use the analysis results	High
TR_82	Data Scientist, Data Engineer	export the results of an analysis (AI pipeline)	I can create presentations and analysis in other tools (e.g. Office, BI tools)	Medium
TR_86	Data Scientist	run automatic tests on the registered AI models (when including them in an AI pipeline)	I can quickly check that the trained models are robust and fault-tolerant	Medium
TR_87	Data Scientist	support the inclusion of different performance metrics as part of an AI pipeline	I can obtain a better picture of my pipelines' / models' effectiveness according to my needs	High

No	As a…	I want to …	So that…	Relevance
TR_93	Data Scientist	have a common metadata model for describing my AI pipelines, experiments, results	I can share them with other users	High
TR_94	Business User	define model evaluation metrics (beyond the ones used by the data scientists)	I can monitor how the model's performance translates to my business KPIs	Medium
TR_96	Data Scientist, Data Engineer	define where my AI pipeline will be executed	I can ensure the analysis is securely executed in infrastructures under my control	Medium
TR_102	Data Scientist, Data Engineer	view detailed execution logs for an AI pipeline	I can detect bottlenecks once my AI pipeline has been executed	Medium
TR_103	Data Scientist, Data Engineer	keep a history of experiments performed (pipeline runs)	I can compare results & improve the pipeline	Medium
TR_104	Data engineer	schedule the execution of the AI pipelines	I can have at my disposal up-to-date results	High
TR_105	Business User, Data Scientist	store the execution results	I can retrieve them to use them in external systems	Low
TR_108	Business user, Data Scientist	execute AI pipelines locally on my private cloud or servers on premise	I can perform my analysis in a secure and trusted environment	Low
TR_111	Data Scientist	retrieve my analysis results through an API	I can use them in the XMANAI manufacturing apps	Medium
TR_112	Data Engineer	containerize the developed AI pipelines	they can be more easily deployed	Medium
TR_129	Data Scientist	compare performances of different AI pipelines	I can track the improvements of the different versions of my pipelines / models	Medium

In addition, the MVP features that are within the scope of the XAI Pipeline Manager are depicted in the following table. It needs to be noted that the bold highlights indicate the MVP features for which the XAI Pipeline Manager is primarily responsible.

Table 4-22: XAI Pipeline Manager - Mapping to MVP Features

ID	Title
XMANAI_F_DM_017	AI Model Management (create/store/update/delete/clone/export/import, configure, versioning, register/import)
XMANAI_F_DM_018	**AI Pipeline Management (create/store/update/delete/clone/export/join, view IPR of assets involved, define templates, versioning)**
XMANAI_F_AI_034	AI Model Training, Application & Evaluation (experimentation vs production, configure control parameters, define/monitor eval . metrics, save check points, support parameter optimisation)
XMANAI_F_AI_035	**AI Pipeline Design (define, configure, register AI model, add annotations/comments, reuse common features)**
XMANAI_F_AI_036	AI Pipeline Execution & Evaluation on XMANAI Common Cloud (experimentation vs production, run automatic tests for AI models, run scheduling, configuration)
XMANAI_F_AI_037	AI Pipeline Execution & Evaluation on Premise / Private Cloud (experimentation vs production, run automatic tests for AI models, run scheduling, configuration)
XMANAI_F_AI_038	**Collaboration over AI pipelines creation (experiments comparison, history of events, simulations of different settings, models, and methods for same task)**
XMANAI_F_AI_039	**AI Pipeline Results Management (store, export, add summary statistics, easily compare with real values (for predictions), retrieve via API)**

ID	Title
XMANAI_F_AI_040	AI Pipeline Results Visualisation (configuration of various charts, add comments, store, export, run on cloud vs on premise)
XMANAI_F_AI_041	AI Pipeline Support (offer recommendations, guidelines, common metadata model, execution logs, error handling, show comp. resources expended)
XMANAI_F_EX_042	Explainability Methods Management (add/remove/configure, register/import)
XMANAI_F_EX_043	**Collaboration over AI model/results/pipelines explanations (application of explainability methods at AI pipeline or model level, results querying)**
XMANAI_F_EX_044	Explainability Results Visualisation (various charts, adjust based on user profile)
XMANAI_F_EXI_045	Explainability Results Evaluation (allow manual feedback & results validation)

4.11.3 Interactions

In terms of expected interactions with other XMANAI components, the XAI Pipeline Manager lies at the core of the XMANAI architecture taking into consideration that, at high-level, it has (or leverages) interfaces with the following components:

- The Data Access Manager for setting the access policies for each XAI Pipeline.
- The Data Manipulation Engine for the configuration of the Data Manipulation tasks in the XAI pipeline.
- The Execution & Orchestration Engine for the execution of a fully configured XAI pipeline.
- The Model Engineering Engine for the configuration of the AI Model tasks in the XAI pipeline.
- The Provenance Engine for tracking the lineage of the data and models leveraged in an XAI pipeline.
- The XAI Marketplace for sharing a fully configured and successfully executed XAI pipeline with other users.
- The XAI Models Catalogue for retrieving the different ML/DL models in collaboration with the Model Engineering Engine.

In detail, the expected interfaces are depicted in the following diagram. It needs to be noted that the XAI Pipeline Manager also interacts with the XAI Visualization Engine that may access the results of an ongoing or finalized XAI pipeline, and the Identity & Authorisation Manager for authentication and authorization, even though the specific interactions are not currently depicted in Figure 4-11.

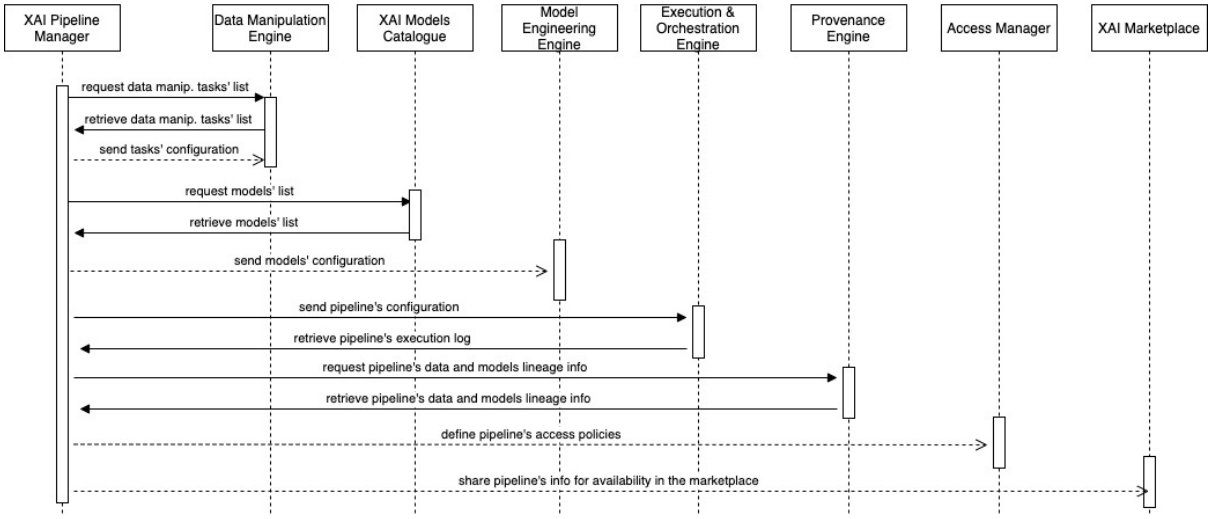

Figure 4-11: XAI Pipeline Manager – Interactions with other XMANAI Components

4.12 XAI Visualisation Engine

The XAI Visualisation Engine constitutes a core component of the AI Insights Services Bundle as it provides the visualisation capabilities of the XMANAI platform encapsulating all offerings of the

platform related to the extraction of meaningful information through multiple visualisation types depending on the user's needs. It is available and served through the Core AI Platform in the XMANAI Centralized Cloud deployment.

4.12.1 Overview

The XAI Visualisation Engine is undertaking the responsibility of providing the novel visualisation capabilities of the XMANAI platform by providing a rich set of modern visualisations which are leveraged by the different components of the XMANAI architecture based on the designed offerings of the platform. To this end, the XAI Visualisation Engine incorporates all visualisation mechanisms that facilitate the generation and exploitation of useful insights, valuable information and new knowledge from the visual representation of the data and their metadata, from the ML models' parameters, metadata and explanatory information, as well as from the experimentation metrics and results.

Hence, the XAI Visualisation Engine effectively covers the visualisation needs across the different components of the XMANAI platform and the different types of users, in different axes through a set of visualisation mechanisms as described below:

- With modern data visualisations that enable the visual representation of the data stored in the platform in order to effectively explore their properties and gain useful intuition that can be leveraged later in the ML and XAI tasks, as required by the Interactive Data Exploration & Experimentation Tool that is described in detail in deliverable D3.1.
- With modern visualisations of the annotations and explanations produced by the utilised explainability tools which will be leveraged to address the explanatory needs during the ML modelling tasks of the platform, as required by the XAI Model Explanations Engine that is described in detail in deliverable D3.1.
- With advanced visualisations of the experimentation results and their metadata that enable the intuitive understanding and comparison of the outcomes of different experiments performed in the XMANAI platform, as required by the Experiment Tracking Engine that is described in detail in deliverable D3.1.
- With state-of-the-art visualisations that facilitate the enhanced and dynamic visual representation of the results generated from the execution of the XAI pipelines towards obtaining a more comprehensive picture of the results and extracting meaningful insights, as required by the Results Visualization Engine that is described in detail in deliverable D3.1.

Depending on the nature of the information that will be visualised, different approaches will be followed, which will allow the use of the XAI Visualisation Engine by all the different users of the platform, e.g. business users and data scientists. Hence, the XAI Visualisation Engine will exploit the most dominant visualisation patterns available in literature and the market, depending on the visualisation needs that need to be addressed. It should be noted that while the XAI Visualisation Engine is considered as a single component in the XMANAI architecture (as explained in Section 2), however under the hood it is composed of various different modules, each one specialised on one of the different axes described above and tightly integrated within the respective components of the XMANAI architecture. Nevertheless, as all these modules are designed in a manner that they are reusable and easily integrated with different components to address the diverse stakeholder needs, the XAI Visualisation Engine is considered as an "umbrella" component of all the visualisation mechanisms of the XMANAI architecture.

The XAI Visualisation Engine consists of different modules that provide the visualization capabilities of: (a) the Interactive Data Exploration & Experimentation, (b) the XAI Model Explanations Engine, (c) the Experiment Tracking Engine and (d) the Results Visualization Engine that are described in detail in the XMANAI Deliverable D3.1.

4.12.2 Mapping to MVP Features & Technical Requirements

The design specifications of the XAI Visualisation Engine were driven by the different technical requirements defined in the XMANAI Deliverable D1.2. As depicted in the following table, the relevance of the related requirements is marked as high (if the XAI Visualisation Engine is mainly responsible for the specific functionality), medium (if the XAI Visualisation Engine contributes to the specific functionality) and low (if the XAI Visualisation Engine mostly leverages the specific functionality from other components in the XMANAI architecture).

Table 4-23: XAI Visualisation Engine - Mapping to Technical Requirements

No	As a…	I want to …	So that…	Relevance
TR_53	Data Scientist	view data distribution/profiling charts or summary statistics for the data (e.g. number of missing values, min/max values)	I can monitor data drifting issues	Medium
TR_83	Data Scientist	add comments to visualisations of AI pipelines	I can inform other team members of [interesting findings, identified errors, inputs/outputs that need to be explored,…]	High
TR_84	Business User	properly visualise the explanations depending on the user profile	I can adapt the information I receive, according to my needs	High
TR_88	Data Scientist	generate multiple visualisations as output of an AI pipeline	I can make it available to the involved users	High
TR_89	Business User	view a visualisation including the results and their explanations	I can take informed decisions	High
TR_90	Business User	choose among different visualisations	I can create the charts and measurements that help me quickly detect the information I want	High
TR_95	Business User	retrieve and compare previous AI model's results with the real outcomes once they are available	I can keep track of the model's performance over time	Medium
TR_103	Data Scientist, Data Engineer	keep a history of experiments performed (pipeline runs)	I can compare results & improve the pipeline	Low
TR_109	Data Scientist	create reports of the performed data analysis locally on my environment	I can evaluate the results	Medium
TR_113	Data Scientist	able to setup a data analysis execution and results visualization environment easily on private cloud or servers on premise	I can leverage the execution of the analysis and visualization of results on infrastructures I control	Medium
TR_128	Data Scientist	compare results from different models created for a particular task	I can gain an understanding of which factors aid in the performance of the models (which features helped, how the different preprocessing steps affected the result	High
TR_129	Data Scientist	compare performances of different AI pipelines	I can track the improvements of the different versions of my pipelines / models	Medium

In addition, the MVP features that are within the scope of the XAI Visualisation Engine are depicted in the following table. It needs to be noted that the bold highlights indicate the MVP features for which the XAI Visualisation Engine is primarily responsible.

Table 4-24: XAI Visualisation Engine - Mapping to MVP Features

ID	Title
XMANAI_F_DP_030	**Data View & Visualisation (query data, view distribution, statistics, aggregations, time periods, descriptive analytics, preview sample)**
XMANAI_F_AI_039	AI Pipeline Results Management (store, export, add summary statistics, easily compare with real values (for predictions), retrieve via API)
XMANAI_F_AI_040	**AI Pipeline Results Visualisation (configuration of various charts, add comments, store, export, run on cloud vs on premise)**
XMANAI_F_AI_041	AI Pipeline Support (offer recommendations, guidelines, common metadata model, execution logs, error handling, show comp. resources expended)
XMANAI_F_EX_043	Collaboration over AI model/results/pipelines explanations (application of explainability methods at AI pipeline or model level, results querying)
XMANAI_F_EX_044	**Explainability Results Visualisation (various charts, adjust based on user profile)**
XMANAI_F_EXI_045	**Explainability Results Evaluation (allow manual feedback & results validation)**

4.12.3 Interactions

As described in section 4.12.1, the XAI Visualisation Engine is providing the visualisation capabilities of the XMANAI platform with a rich set of modern visualisation techniques, which are leveraged by the different components of the platform based on their needs. Hence, the XAI Visualisation Engine interacts with the components that require the proper and effective visualisation of the data stored in the platform and their metadata, the parameters, metadata and explanatory information produced in XAI ML training, pipeline design and execution phase, as well as of the produced results and their comparison during the experimentation phase. To achieve this, the XAI Visualisation Engine provides a set of well-defined APIs in order for the rest of the components to be able to utilise the offered visualisation mechanisms. To this end, the XAI Visualisation Engine mainly interacts via the respective endpoints with the following components:

- The Data Manipulation Engine Tool for the data visualisation of the underlying data which are stored in the platform's storage as well as their metadata.
- The Model Engineering Engine for the visualisation and presentation of the explanations that are generated as well as their metadata, as well as for facilitating the visual presentation and comparison of the produced experimentation results and their metadata.

In detail, the expected interfaces are depicted in the following diagram. It needs to be noted that the XAI Visualisation Engine incorporates the visualisation mechanisms of the platform hence the following diagram illustrates the main interactions of the XAI Visualisation Engine with components that utilise its visualisation mechanisms. In particular, the XAI Visualisation Engine also interacts with the Data Access Manager, and the Identity & Authorisation Manager, even though the specific interactions are not currently depicted in Figure 4-12.

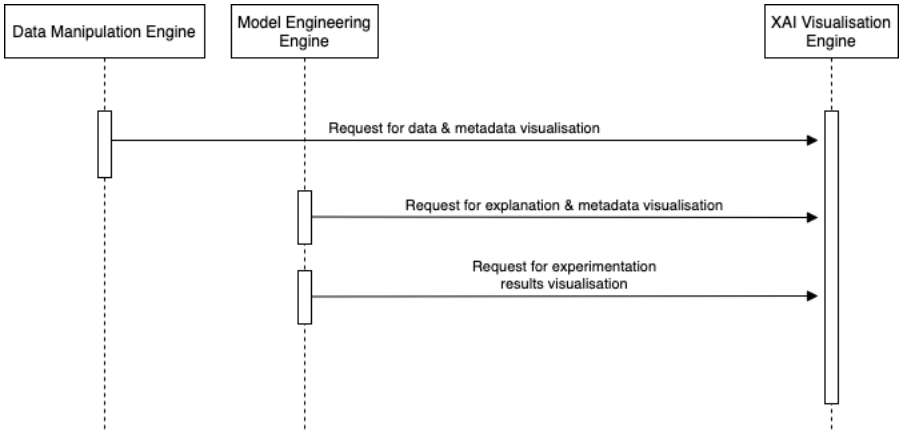

Figure 4-12: XAI Visualisation Engine – Interactions with other XMANAI Components

5 XMANAI Manufacturing Apps

This section outlines the design of the XMANAI Manufacturing Applications in terms of functionalities, mapping to the XMANAI business requirements and anticipated interactions with the XMANAI Platform.

5.1 Process Optimization App

The current situation at Ford Engine Plant does not allow the power of integrated data to be harnessed for decision making. There are records on the status of the different operations on the production line, the quantity of engines produced and their parts, quality reports and production plans. Despite having this information, there is not a centralized database and all the information is disaggregated in different corporate databases. This lack of centralized information is the first problem that needs to be solved in order to optimize the different processes that occur on the production line. This problem implies another one, which is the lack of artificial intelligence applied to the different decision-making processes due to the impossibility of taking advantage of all the available data. The proposed application aims to mitigate these problems by means of a set of functionalities that will be explained in the following section.

5.1.1 Overview

The application expected to be developed within the project provides two main functionalities: the production overview and the automated production planning.

The production overview is composed of different components. First of all, different data sources related to production data will be joined to represent the status of the production line at any moment and to make predictions about the amount of engines produced at the end of the shift following the current trend of the line. Both information will help the business experts to understand the significant deviations that may occur between the expected (planned) production and the actual engines produced at the end of the shift. In addition, the application will provide an alert system for anomalous situations that will occur in case the production line behaves according to a certain trend without actions being taken to reverse this situation. The last component focuses on the simulation of different scenarios (changes at some point of the production line) to analyse the impact of these changes on the estimated production at the end of the shift.

The automated production planning part aims to solve different problems related to the current way of planning production, which may not be optimal due to the lack of capacity to process and take into account the information about the actual status of the production line as well as the demand of engines in the different working shifts. Thus, this part of the application will focus on the generation of constraints for production scheduling based on the current status of the production line. In addition, a component will be developed for the sequencing and planning of the production of the different engines during a shift, replacing the current tool with one that takes into account both the demand and the previous generated constraints. Finally, it is intended to use the application to suggest automated daily planning for a whole month besides those proposed by the business experts, taking advantage of all available data sources.

5.1.2 Mapping to Business Requirements

The following table presents the different business requirements and explains how they are expected to be addressed by the app's functionalities. The following requirements are considered as outdated and have not been included in the table: "XMANAI shall send advice based on real time data to help to take actions about the stocks.", "XMANAI should alert for critical parts, that sufficient stocks of some parts to finish the plan are not available."

Table 5-1: Process Optimization App - Mapping to Business Requirements

BUSINESS REQUIREMENTS	XMANAI APP FUNCTIONALITIES
When a failure affect occurs, XMANAI shall represent in real time the machine/part related to the failure.	By means of real-time representation of the line status, the application will provide the necessary information to detect the different parts of the line involved in a failure.
XMANAI should integrate all corporative data to know the status of machines in real time.	The app will present useful information and insights from different corporate databases that are collected in the XMANAI platform.
XMANAI should allow operators and engineers to visualize the data from any locations and from multiple devices (smartphone, laptops).	The application will be ready to be viewed on multiple devices.
XMANAI should save historical data to help engineers to review the historical actions.	The processing of data from corporate databases in the XMANAI platform will allow the users to view historical data in the XMANAI app.
XMANAI shall show direct and visual alarms to alert about critical situations.	The anomalous situation alert system of the app (triggered through the XMANAI platform) will take care of this requirement.
XMANAI should allow an operator to understand the root causes in every working situation.	Through the explainability functionalities provided by the XMANAI platform, an operator will be able to understand the predictions provided by the different components of the app.
XMANAI should have flexibility to be applied in different lines.	Although in XMANAI the application will cover one line, it is expected to be flexible to extend its use to other lines with little effort.
XMANAI should provide advices in terms of production plan.	The app will provide different automated plannings.
XMANAI should read and integrate data from corporate databases and external sources.	The app will present useful information and insights from different corporate databases that are collected in the XMANAI platform.
XMANAI should act as simulator of different planning scenarios.	Using the explanations provided by the automated planning model, the application will simulate different planning scenarios based on a specific manual action included in the planning.
XMANAI should consider the production that is currently in the line but it hasn't yet finished.	This production will be taken into account in the planning and sequencing of engines during a shift.
XMANAI should be agile to replan when an unexpected event occurs in the line.	Through the constraints generated from the line status information, a new shift planning and sequencing would be produced.
XMANAI should provide measurements of the deviation between the predicted plan and the real production.	This will be part of the information collected and calculated by the XMANAI platform and made available through the app.

5.1.3 Interactions

The XMANAI platform will connect to the Ford engine plant to collect new data from the corporate databases. This data will feed the different XAI pipelines designed to cover the different functionalities described in the Overview section. These pipelines will be executed to send the necessary data, predictions and associated explanations to the app, which will display this information as expected. Business experts will take advantage of the added value provided by the app to make the decisions they consider necessary based on the available information.

5.2 Product Demand Forecasting App

As of today, the Whirlpool relatively new B2C channel, targeted at end users rather than selling wholesale to other retailers, does not have a specific forecasting process but follows the overall B2B demand forecasting for the whole Whirlpool EMEA. This leads to a poor and inaccurate forecasting estimation, that gets worse when considering the high complexity of consumers' behaviour and the recent history of sales over consumers, leading to poor historical data availability. Due to these considerations, within XMANAI, the demonstrator aims at developing a proper B2C forecasting application for accurate consumers' sales prediction based on explainable Artificial Intelligence.

5.2.1 Overview

The foreseen application will allow to focus on B2C forecast, better predict consumers demand though the use of more complex and complete algorithms provided by XMANAI. In particular, the application will generate forecasts on a daily basis to cover a time horizon of 3 months (13 weeks) and it allow the WHR internal planner to have an additional mean to inspect the forecast proposal leveraging on a proper dashboard with Explainable AI outputs in different forms. Indicatively, heatmaps shall be generated combining geographical tags with numerical parameters, illustrating the concentration of that parameter across a geographical region of interest and will be used to allow WHR internal planner for an immediate interpretation of the achieved results. In particular, the foreseen heatmaps will represent: (1) the concentration of registered users in Europe, (2) the density of session time in Europe and (3) the geographic distribution of the owner's Revenue.
The B2C demand profile will be visible to the central planner within the XMANAI platform and the app, and it will be then aggregated with the B2B channel demand, by manually injecting it into the IBP Whirlpool process.
The demand profile will also be used for B2C inventory management purposes and, in the midterm, to drive supply base decisions for the final distribution. The app will give the possibility to all process stakeholders to understand the reasons and key factors leading the achieved results.
Moreover, additional stakeholders having access to the app, will have the possibility to simulate some key factors modification (i.e., supply issues, sales actions, price modification, range definition), having a real time recalculation of the demand profile.
The application will eventually make visible sales initiatives opportunities providing suggestion and modification.

5.2.2 Mapping to Business Requirements

The following table presents the different business requirements and explains how they are expected to be addressed by the app's functionalities.

Table 5-2: Product Demand Forecasting App - Mapping to Business Requirements

BUSINESS REQUIREMENTS	XMANAI APP FUNCTIONALITIES
XMANAI should allow a central planner to have a correct forecasting of D2C sales per day/product in a horizon of 3 months.	XMANAI Platform will provide complex and accurate algorithms for D2C predictions, generating D2C demand on a daily basis to cover 3 months time horizon. The results of such algorithms will be presented in the app.
XMANAI should generate demand forecasts on a daily basis.	
XMANAI shall allow central planners/D2C marketing, D2C sales/D2C logistics to visualise the key factors effects influencing demand profile.	The envisaged heatmap and other visual charts offered in the app dashboard will give the possibility to all process stakeholders to understand the reasons and key factors leading the achieved results.
XMANAI shall allow central planners/D2C marketing, D2C sales/D2C logistics to see the clustering of customer behaviour.	Heatmaps will allow, through an immediate visual representation, to see the consumers behaviours

BUSINESS REQUIREMENTS	XMANAI APP FUNCTIONALITIES
XMANAI shall allow central planners/D2C marketing, D2C sales/D2C logistics to visualise buying patterns per customer profile/product/period.	distributed around Europe, based on results retrieved from the XMANAI platform.
XMANAI shall allow D2C marketing/D2C sales to receive recommendations/input for promotional actions.	The application will eventually make visible sales initiatives opportunities providing suggestions and modifications.
XMANAI shall allow central planners/D2C marketing, D2C sales/D2C logistics to simulate demand forecasting forcing the change in one or more key parameters.	Additional stakeholders having access to the app, will have the possibility to simulate some key factors modification (i.e., supply issues, sales actions, price modification, range definition), having a real time recalculation of the demand profile.
XMANAI shall strictly authorize users for system access.	Access policies will be properly defined in the XMANAI platform and enforced in the app.
XMANAI shall strictly protect all sales data through encryption and secure data management, in compliance with Whirlpool data security policies.	Security access processes will be properly defined and enforced in the XMANAI private cloud deployment.
XMANAI shall provide full customer data anonymization.	Customer data will be anonymized.
XMANAI shall fully respect GDPR (General Data Protection Regulation) by a privacy by design approach, in compliance with Whirlpool data security policies.	GDPR policy will be implemented.

5.2.3 Interactions

The XMANAI Platform will provide complex and accurate algorithms for D2C predictions, generating D2C demand on a daily basis to cover 3 months ahead time horizon. The demand profile will be visible to the WHR central planner within the XMANAI platform from where it will be downloaded to be injected manually into the IBP Whirlpool process, where it will be merged with the traditional B2B demand. All the necessary predictions and explanations will be created in the XMANAI Platform and displayed in the Product Demand Forecasting app (as explained in Section 3.5).

5.3 Process/Product Quality Optimization App

In the CNH Modena production plant, manufacturing equipment is often utilized without a planned maintenance approach and this affects production lines causing sudden and undesired stoppages of the machines, resulting in availability and performance losses.

The use of Artificial Intelligence could radically support maintenance operations, and in particular those activities that deal with the diagnostics of electronic components, that are currently considered as “black box” for CNH maintenance operators. Nowadays maintenance activities on electronic components are indeed performed randomly, by substituting the specific components according to the available spare parts or trying to guess what happened. The introduction of an AI-based system would support operators’ intervention on unexpected downtimes through an advanced troubleshooting system combined with Augmented Reality.

5.3.1 Overview

The application that will be developed for CNH is a collaborative troubleshooting and root cause analysis system, combined with Augmented Reality to support operators in executing maintenance activities on CNH Heller400 machines based on data and insights retrieved from the XMANAI platform.

The machine will be equipped with an AI-based system able to collect data from specific sensors about the status of the machine and the connected electronic components. When a fault/error occurs on the machine, an alarm is generated and made visible through specific graphic user interfaces to the operator, that can follow the suggested procedures and/or start the troubleshooting. Based on the

root-cause analysis, the AI system will begin the troubleshooting defining a list of possible procedures to be followed and correspondent material/components to be substituted and explaining to the operator the reason behind a specific suggested action and why the fault/error occurs.

Moreover, the troubleshooting application will be combined with Augmented Reality functionalities, that has proven its potential and are considered highly effective also for training new maintenance recruits and for supporting not experienced operators. The human operator will be equipped with a suitable device (tablet/smartphone/smart glasses) that is connected with the AI system, receives information about the alarm and shows the operator the procedures to be followed.

The Augmented Reality system will be also used for training new operators in performing repetitive maintenance activities, such as first machine reactivation, that can occur several times and on different machines.

5.3.2 Mapping to Business Requirements

The following table presents the different business requirements and explains how they are expected to be addressed by the app's functionalities.

Table 5-3: Process/Product Quality Optimization App - Mapping to Business Requirements

BUSINESS REQUIREMENTS	XMANAI APP FUNCTIONALITIES
XMANAI should provide an interactable digital twin able to forecast the behaviour of the machinery.	The models and pipelines to be used in the expected application will be trained in the XMANAI platform in order to eventually forecast the machinery behaviour and alert operator on time before error occurs, suggesting possible operations/substitutions to be done.
XMANAI should provide alarms with description of the problem and visibility on how the problem has been forecasted.	When a fault/error occurs, an alarm is generated and made visible to the operator in the shopfloor through specific graphic user interfaces which explain why the fault/error occurs and what happened.
XMANAI should provide an interactable HMI able to improve the comprehension of the suggestion provided by the XAI and to navigate them.	Proper GUI/HMI will be developed and made available on the machine and on the smart device (tablet/smartphone/smart glasses) of the operator.
XMANAI should provide a set of data to the user identifying the problem before the critical moment, the maintenance/troubleshooting procedure to be executed and the parameters to be monitored during production.	Appropriate XAI models will be trained in the platform to provide scheduling suggestions and information to the operators on maintenance activities to be performed. This information and procedures will be made visible to the operator through the GUI/HMI.
XMANAI should support the Blue Collar Worker is doing the maintenance/troubleshooting procedures with AR/XAI connection.	The troubleshooting application will be combined with Augmented Reality functionalities, by equipping the operators with smart devices.
XMANAI should provide an interactable HMI able to navigate data from different devices including PC, tablet/smartphone and wearable AR devices .	

5.3.3 Interactions

XMANAI platform will mainly connect the Modena Plant Shopfloor and the Digital plant containing the factory systems. The platform will firstly receive raw data coming from the sensor installed in the machine through proper data pipelines. This data will be ingested by specific XAI algorithms and novel AI models supporting the operator in data evaluation with explainable AI through the use of different visualization tools. Based on data evaluation, the platform will initiate the root cause analysis and suggest maintenance operations to be followed. In particular, the human operator will be equipped with an AR system receiving information from the XMANAI platform, thus visualizing and understanding data directly in the shopfloor.

5.4 Process Optimization & Semi-Autonomous Planning App

Optical technology is often used for dimensional control in automotive, aeronautical or energy sectors, for instance. Compared to traditional measurement methods, optical measurement systems allow a fast acquisition of huge amounts of data. One of the limitations of this technology is that expertise is needed; depending on the person running the measurements, the results may differ.

The use of Artificial Intelligence will allow the development of a recommender that guarantees the quality of the results obtained, evaluating and certifying the algorithms used by the measurement software and eliminating the “black box” that nowadays exists.

5.4.1 Overview

The Process Optimization & Semi-Autonomous Planning (or Explainable Metrology 4.0) app will be developed to improve UNIMETRIK M3 Software functionalities through bilateral interaction with the XMANAI platform. The application will speed up the measurement process because optimal parameters will be used from the beginning based on AI results acquired from the XMANAI platform.

The measurement machine is already equipped with a sensor. Data collected by the machine software will be communicated to the application. First, the data will be filtered and only relevant points will be saved. That is, data out of range, measurement errors and others are eliminated to obtain an optimal point cloud. In addition, after data analysis and algorithm training, the program will be able to recommend optimal measurement plan parameters and represent the precision that will be achieved on the measurement. This functionality will reduce the variation between measurements due to operator expertise.

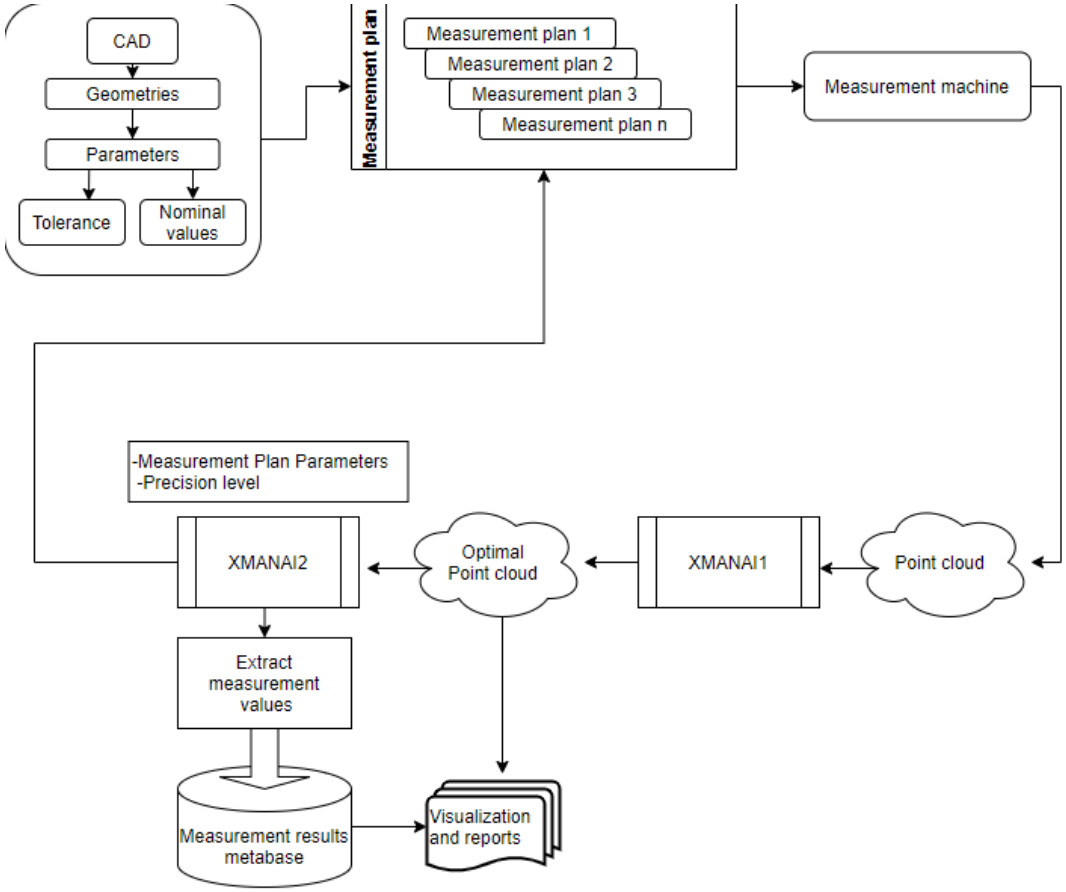

Figure 5-1: Process Optimization & Semi-Autonomous Planning App Interactions

In more detail, the data retrieved from a measurement machine will be processed by an XMANAI pipeline (XMANAI1 in the figure) to generate an optimal point cloud eliminating not relevant points,

and thus, reducing the dataset to be handled. This step will accelerate the parameterization of the measurement plans. In addition, based on data evaluation, another pipeline (XMANAI2 in the figure) will recommend the optimal measurement plan parameters along with the level of precision with which the measurement will be achieved.

5.4.2 Mapping to Business Requirements

The specific app on Explainable Metrology 4.0 is designed as a layer on top of the UNIMETRIK M3 Software, rather than as a stand-alone app, taking into consideration different business requirements defined in the XMANAI Deliverable D1.2.

Table 5-4: Process Optimization & Semi-Autonomous Planning App - Mapping to Business Requirements

BUSINESS REQUIREMENTS	XMANAI APP FUNCTIONALITIES
XMANAI shall be able to adjust the displayed criteria based on the geometry type and allows full control to add various GD&T checks and other specific location information for an element.	The XAI pipelines executed in the XMANAI platform shall be able to return optimal parameters from the data feed entered in an AI model (lateral density, scan direction density, exposure time) with precision level indicator (low, medium, high). The results will be displayed in the UNIMETRIK M3 Software through the app.
XMANAI shall be able to automatically add and connect to an instrument using predefined parameters without any user interaction after configuration starts.	
XMANAI shall reduce the amount of interaction with the software so that users spend more time measuring and less time browsing through the software.	The XMANAI platform will provide operators regardless of their level of experience with a clear representation of the optimal point cloud results obtained in the model.
XMANAI should collect and standardize machining data.	The app (and indirectly the UNIMETRIK M3 Software) will retrieve standardized data complying with the XMANAI data model through the XMANAI platform.
XMANAI shall be able to provide information to detect easily the sources of problems.	The AI model created in XMANAI shall generate an optimal point cloud and be self-learning in capturing possible errors in order to avoid not relevant points.
XMANAI shall create pop-up messages for user instructions, instrument alignment, profile change, etc.	The platform shall have a large database from which to perform self-learning in order to give optimal results for the metrology objects to be analysed.
XMANAI shall keep historical records of the machine on-site.	
XMANAI shall be able to analyze and communicate results.	The app will report a visualisation map about the optimal parameters of the object to be analysed.
XMANAI shall create predefined measurement routines and explain why they are the correct ones.	The AI models will be trained and will provide an optimal measurement plan.
XMANAI shall be able to reduce the number of iterations required with the computer by the user, so the data collection will be more efficient.	
XMANAI shall perform improved visualizations of item annotations and value logic.	Additional GUI/HMI will be developed as needed and made available for the operator

5.4.3 Interactions

The XMANAI Platform will be connected to a UNIMETRIK metrological software. The platform will receive the data from the measurement machine and utilize it in the execution of the appropriate XAI pipelines. The results of the XAI pipelines will be displayed to end users through the Process Optimization & Semi-Autonomous Planning app (that is positioned on top of existing software in UNIMETRIK).

6 Conclusions and Next Steps

The present deliverable (D5.1) documents the XMANAI reference architecture as the main outcome of Task T5.1 with three main objectives: (a) to elaborate on the architectural blueprints from different perspectives and abstraction layers, (b) to define the core workflows that require interaction among the different XMANAI components to deliver the intended functionality for AI preparation, experimentation, insights extraction and application, and (c) to provide the initial specification of the XMANAI components and manufacturing apps that shall guide the impending design and development activities. To derive these outcomes, a clear approach was followed, based on intense collaboration among all technical partners and involving multiple iterations to brainstorm on the functionalities of the different components and services.

The XMANAI reference architecture has leveraged the key findings of tasks T1.3 “Platform Requirements Elicitation, Data Acquisition and AI Scenarios” and T1.4 “XMANAI Concept Elaboration, MVP Definition and Validation” as reported in the XMANAI Deliverable D1.2 (“XMANAI Concept Detailing, Initial Requirements, Usage Scenarios and Draft MVP”), as well as the business requirements elicited in T6.1 “Demonstrators Requirements Elicitation” and presented in the XMANAI Deliverable D6.1 (“Demonstrators Requirements”). In fact, following a user-driven approach, the business and technical requirements, as well as the prioritized MVP features, have been instrumental in defining the functionalities of the different XMANAI components and manufacturing apps. In addition, the business user journey, the data scientist journey and the data engineer journey (that were elaborated in D1.2) have been revisited in order to formulate the basic platform workflows and reveal the logical interactions between the different components.

The XMANAI reference architecture is expected to be refined and further detailed in the future while the updates performed will be reported with the different XMANAI platform releases on M21 (alpha release), M32 (beta release) and M42 (Release 1.0) that shall be documented in D5.2, D5.3 and D5.4, respectively. Since the T5.1 activities remain active till M42, the next steps along the proposed work involve the take-up of the architecture design by: (a) the Data & AI Services Bundles in WP2 “Industrial Asset Management and Secure Asset Sharing Bundles” and WP3 “Core Artificial Intelligence Bundles for Algorithm Lifecycle Management” since the components are further specified in their design and development activities, (b) the AI Models-related tasks of WP4 “Novel Artificial Intelligence Algorithms for Industrial Data Insights Generation” with regard to the implementation of the draft XMANAI XAI Models Catalogue, (c) the manufacturing apps development activities per demonstrator within WP6 “Demonstrators Setup, Operation and Business Value Exploration”, and (d) the development, integration and deployment activities to be performed in WP5 “XMANAI Platform Continuous Integration”.

References

XMANAI Description of Action (DoA), 2020.

XMANAI Deliverable D1.1 “State-of-the Art Review in XMANAI Research Domains”, 2021.

XMANAI Deliverable D1.2 “XMANAI Concept Detailing, Initial Requirements, Usage Scenarios and Draft MVP”, 2021.

XMANAI Deliverable D2.1 “Asset Management Methods and System Designs”, 2021.

XMANAI Deliverable D3.1 “AI Bundles Methods and System Designs”, 2021.

XMANAI Deliverable D6.1 “Demonstrators Requirements”, 2021.

Industrial Internet Consortium (2019). The Industrial Internet of Things, Volume G1: Reference Architecture. Version 1.9. Retrieved on September 9th, 2021 from https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf

RAMI 4.0. Retrieved on September 9th, 2021 from https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.pdf?__blob=publicationFile&v=7

List of Acronyms/Abbreviations

Acronym/ Abbreviation	Description
AI	Artificial Intelligence
BR	Business Requirement
DoA	Description of Action
IIRA	Industrial Internet Reference Architecture
MVP	Minimum Viable Product
RAMI 4.0	Reference Architectural Model for Industrie 4.0
TR	Technical Requirement
WP	Work Package
XAI	Explainable Artificial Intelligence
XMANAI-PrOp	XMANAI Production Optimization App
XMANAI-PDeF	XMANAI Product Demand Forecasting App
XMANAI-PPQO	XMANAI Process/Product Quality Optimization App
XMANAI-SAMP	XMANAI Process Optimization & Semi-Autonomous Planning App