A Holistic Digital Mine 4.0 Ecosystem

D2.4
Mine.io architecture specification - version 2

Editor(s)	Jan Niemi (LTU), Dimitris Kassimis (INN), George Kouzas (INN), Lisa Erba (GFT)
Lead Beneficiary	LTU
Status	Final
Version	1.0
Due Date	30/09/2024
Delivery Date	30/09/2024

Dissemination Level	PU - Public

MINE.IO is a project co-funded by the European Commission under the Horizon Europe Programme under Grant Agreement No. **101091885**

Project	MINE.IO – 101091885
Work Package	WP2 – Understanding the lifecycle Mine.io digitalization
Deliverable	D2.4 - Mine.io architecture specification - version 2
Contributor(s)	Max Friedemann (TUBAF), Vassilios Protonotarios (AMDC), Paul Schönherr (TUBAF), Raul Mollehuara Canales (UOULU), Wojciech Anacki (KGHM), Jacek Galas (LITR), Angelina Despotopoulou (FRON)
Reviewer(s)	Raul Mollehuara Canales (UOULU) Vasileios Protonotarios (AMDC)

Abstract	This deliverable introduces the architecture of the Mine.io platform based on the 4+1 views methodology. The presentation of the architecture includes the specification of the main technical components of the project's platform, including the information flows and interfaces among them. The above analysis is reflected in the logical, process, implementation, deployment and Use Case views of the Mine.io architecture. Furthermore, the deliverable validates this architecture against the mining scenarios of the project's Pilots.

Disclaimer	The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Communities. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein. © Copyright in this document remains vested with the MINE.IO Partners

HISTORY OF CHANGES

Version	Date	Changes/Content	Partner
V0.1	26/05/2023	Initial Table of Contents for discussion in architecture (T2.5) telco	INN
V0.12	22/06/2023	Introductory Section	INN
V0.14	26/06/2023	Updates on the Methodology Employed	INN
V0.15	19/07/2023	Contribution on the Reference Architectures used for Digital Mines and Related Industries	INN
V0.16	11/08/2023	Contribution on CI/CD Infrastructure	GFT
V0.17	22/08/2023	Updates on Implementation and Development Views	GFT
V0.18	01/09/2023	Contribution on the Requirements from Mine.io Grant Agreement, Use Cases and Inventory List	INN
V0.20	08/09/2023	Contribution on Scenarios-Based Architectures and Process View diagrams for the Use Cases	Technical Partners
V0.40	12/09/2023	Contributions on deployment details for the major component	Technical Partners
V0.50	15/09/2023	Pre-final version ready for consortium and peer review after extensive WP Leader technical review	GFT INN
V0.70	19/09/2023	Integration of the comments received from consortium partners and peer reviewers	GFT INN
V0.90	22/09/2023	Final deliverable review by WP2 Leader	GFT INN
V1.00	27/09/2023	Project Coordinator review and online submission	GFT
V2.01	2/09/2024	First draft of the second version (V2) of deliverable D2.4	LTU
V2.10	11/09/2024	Contributions on architecture	INN
V2.15	22/09/2024	Integration of the comments received from consortium partners and peer reviewers	LTU
V1.0	30/09/2024	Project Coordinator review and online submission	GFT

EXECUTIVE SUMMARY

One of the main objectives of the Mine.io project is to provide a structured approach for the development and deployment of digitally enabled mining solutions. In this direction, the project is developing an architecture for digital mining applications, which provides the structuring principles that drive the integration of different Mine.io technological solutions in end-to-end mining application. The present deliverable presents the current state of the Mine.io architecture based on the 4+1 Views Methodology, i.e. description of the Mine.io architecture in terms of its logical, process, implementation and deployment view, as well as its Use Cases.

From a methodological viewpoint, the specification of the Mine.io architecture has considered State-of-The-Art reference architectures for the mining industry, the requirements of the Mine.io Use Cases and Pilots, as well as the capabilities and deployment needs of the Mine.io technologies. Based on these considerations, the deliverable presents a logical view of the Mine.io architecture, which details the main components of the architecture and their functionalities, as well as some of the interfaces between them.

Furthermore, the specification of the architecture has been the outcome of a top-down approach that considered reference architectures and blueprints to specify a suitable architecture for the Mine.io needs and of a bottom-up synthesis approach that has considered the low-level requirements of the Mine.io technologies and use cases to synthesize a suitable architecture. These two approaches have converged to the outcomes presented in this deliverable.

In detail, different reference architectures have been considered and analysed, including Industrial Internet Reference Architecture (IIRA), Reference Architecture Model Industrie 4.0 (RAMI 4.0), Cisco Industrial Automation for Mining – Cisco Reference Design (CRD), Rockwell Architecture for Mining, as well as architectures from relevant EU projects, i.e. Arrowhead and FIWARE. These architectures propose a layered structure which have influenced the design of the Mine.io architecture.

Regarding the bottom-up approach, also considered and coupled with the above-mentioned top-down review of similar reference architectures, important input has been obtained by analysing the requirements specified for each Mine.io component or integrated solution. This information has been retrieved by breaking down the functionalities, technical specifications and integration issues for the implementation of the components, the analysis of the Use Case scenarios and the requirements of the software/hardware components to be utilised.

Overall, the deliverable provides:

- A detailed presentation of the logical view of the Mine.io architecture where the main operations of the mining stages have been included, along with additional support technologies, integration and communication layers, and device-interfacing protocols.
- A detailed list of the Mine.io technologies which are structured in an inventory and have been considered in generalizing the Mine.io architecture.
- A set of process views for the architecture of the Mine.io pilot systems, which provide information on the data flows across the different components that they comprise.
- Initial insights about the implementation and deployment of the architecture in-line with modern agile CI/CD (Continuous Integration/Continuous Deployment) approaches to integrating software systems.
- A comprehensive description of the project's IT infrastructure that will support the integration of Mine.io systems and Use Cases in-line with the presented architecture.

This deliverable presents the current state of the Mine.io architecture in-line with the 4+1 methodology. It is an update to deliverable D2.2, incorporating the work done in WP5 related to the Mine.io architecture, the development of Mine.io use cases in WP6, and the development efforts by Mine.io's technical partners.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TERMS AND ABBREVIATIONS

3D	Three-Dimensional
ACT	Advanced Control Technology
AI	Artificial Intelligence
API	Application Programming Interface
AUV	Autonomous Underwater Vehicle
AWS	Amazon Web Services
CAD	Computer-Aided Design
CI/CD	Continuous Integration/Continuous Deployment
CPwE	Converged Plantwide Ethernet
CPS	Cyber-Physical System
CRD	Cisco Reference Design
CRM	Critical Raw Materials
CT	Computed Tomography
DDBB	Document-Oriented Distributed Database
DDS	Data Distribution Service
DDL	Deep Learning
DLT	Distributed Ledger Technologies
DNS	Domain Name System
DoA	Description of Action
DT	Digital Twin
DTLS/TLS	Datagram Transport Layer Security
DTM	Digital Terrain Model
EC	European Commission
ERI	Electric Resistivity Imaging
ESG	Environmental, Social and Governance
FMCW	Frequency Modulated Continuous Wave
FTIR	Fourier Transform Infrared Spectrometry
GA	Grant Agreement
GEE	Google Earth Engine
HLRA	High-Level Reference Architecture

HSC	H (enthalpy), S (entropy) and Cp (heat capacity).
IAM	Identify Access Management
IEEE	Institute of Electrical and Electronics Engineers
IDEs	Integrated Development Environments
IIRA	Industrial Internet Reference Architecture
IIoT	Industrial Internet of Things
IoT	Internet of Things
IPT	Inductive Power Transfer
IR	Infra-Red (spectrum)
ISO	International Standardization Organization
IT	Information Technology
IWaas	Infinitech Way as a service
JSON	JavaScript Object Notation
K8s	Kubernetes
LDAP	Lightweight Directory Access Protocol
LCA	Life-Cycle Assessment
LCDN	Life Cycle Data Network
LCI	Life Cycle Inventory
LIBS	Laser-Induced Breakdown Spectroscopy
LiDAR	Light Detection and Ranging
ML	Machine Learning
MQTT	Message Queue Telemetry Transport
MWD	Measurement While Drilling
OBD	On-Board Diagnostics
OEE	Overall Equipment Effectiveness
OMG	Object Management Group
OPC-UA	Open Platform Communications Unified Architecture
OSD	On-Screen Display
OT	Operational Technology
PCL	Passive Coherent Location
PIT	Photonic IT

PLC	Programmable Logic Controllers
PLM	Product Lifecycle Management
RA	Reference Architecture
RAMI	Reference Architecture Model Industry 4.0
RDBS	Relational Database System
REST	Representational State Transfer
RGB	Red-Green-Blue
SCADA	Supervisory Control and Data Acquisition
SDM	Simulation Data Management
SIA	Smart Industry Architecture
SoTA	State-of-The-Art
SOA	Service-Oriented Architecture
TCP	Transmission Control Protocol
UAV	Unmanned Aerial Vehicle
UC	Use Case
UDP	User Datagram Protocol
VDSC	Vehicle Detection Control System
VIS	Visible (spectrum)
WAXS	Wide Angle X-ray Scattering
WiFi	Wireless Fidelity
XRD	X-Ray Diffraction
XRF	X-Ray Fluorescence
XRT	X-Ray Transmission

1. INTRODUCTION

1.1 PURPOSE AND SCOPE

Mine.io is developing, validating, and demonstrating the vision of the future digital mine. In this direction, the project designs and develops technologies and processes for the digital transformation of the mining industry. These technologies and processes are validated in the scope of real-life mining operations (e.g., exploration, extractions, mine-site restoration) in many different use cases across six different pilots' sites. Apart from developing individual technologies and deploying mining applications in the various sites, the project aspires to develop a platform that can support the digital transformation of mining processes based on a host of technological components that work synergistically. Therefore, the project is also concerned with the integration of different technologies in end-to-end mining solutions that digitize and support a variety of mining processes. This requires the development of structuring principles and blueprints for integrating different technological components into complete digital solutions for the mining industry. In the scope of Mine.io these structuring principles are documented and provided as part of the Mine.io architecture.

The Mine.io architecture, presented in detail through its 4+1 View in chapter 5 (logical, process, implementation and deployment views) and chapter 6 (Use Cases views), illustrates how different technologies for the mining industry can be combined in an integrated digital solution. It specifies a variety of components that comprise digital mining solutions along with the information flows and interfaces between them. Most importantly, it provides the structuring principles for combining different technologies in end-to-end solutions. Moreover, it provides insights on how to implement and to deploy such solutions in mining sites, including information about the implementation technologies and the deployment patterns of integrated solutions. As such the Mine.io architecture plays a key role in integrating the Mine.io technologies and in deploying integrating solutions in the pilot sites.

In this context, the present deliverable is devoted to the introduction and description of the Mine.io architecture. The current deliverable is the second version of Mine.io architecture and focuses on the architecture's functional aspects, initial implementation, and deployment considerations. One more release of the present deliverable will be produced till the end of the project, which shall elaborate on the implementation and deployment aspects of the architecture, while at the same time providing best practices and blueprints for building practical digital mining solutions based on the architecture.

1.2 Deliverable Methodology

A variety of methodological instruments and tools have been employed to develop the present deliverable, as outlined in the following subsections.

1.2.1 4+1 Views Methodology

The present deliverable leverages the popular 4+1 View methodology [Kruchten95] for describing software architectures. This is because the Mine.io platform is largely a collection of software and middleware components, which are integrated into solutions based on the Mine.io architecture. According to the 4+1 methodology, the different aspects of a software system can be described as a collection of complementary views including (see Figure 1):

- A logical view that focuses on the functional aspects of the system and represents the view of the end-users.
- A process view that focuses on the information flows and interactions between different components of the system when used in the context of some use case. It is a view of

primary interest to system integrators and addresses aspects like performance and scalability.
- An implementation view that describes how the system is mapped to software components that implement it, including mappings to software packages and libraries. It is a view for programmers and developers.
- A deployment view that describes how the system is deployed. It covers networking and communication aspects, as well as system topology aspects. It addresses the architecture from a system engineering viewpoint.
- A use cases view that illustrates how the system supports the various use cases that the system is destined to support.

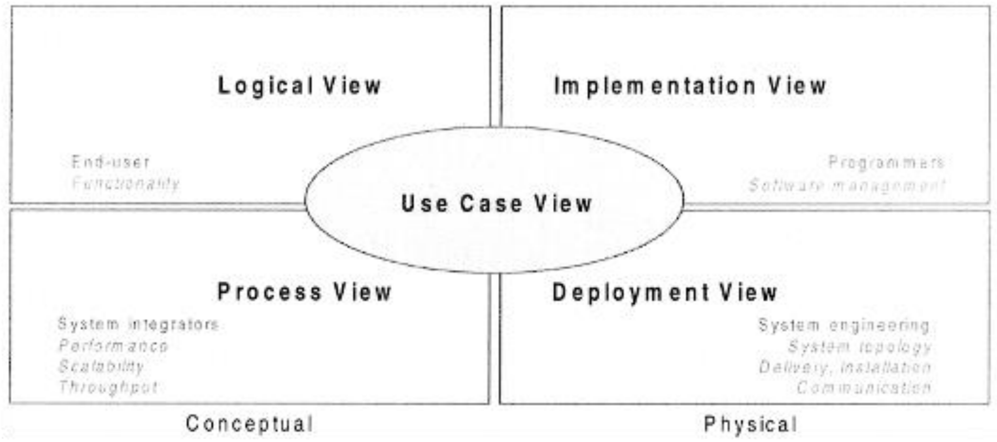

Figure 1: Overview of the 4+1 Methodology for Developing and Documenting Architectures of Software Systems

Mine.io considers the above-listed views essential to different stakeholders of the project's digital mining platform and ecosystem. In addition to offering good coverage and completeness in the presentation of the Mine.io architecture, the 4+1 views methodology provides the means for breaking down the complexity of the architecture description task into different phases. The first version of the architecture deliverable focused on the logical view of the architecture, while providing preliminary insights and considerations for the other views. This version of the deliverable shows the current state of the Mine.io architecture. In the version 3 of the deliverable the final Mine.io architecture will be presented, as illustrated in Figure 2.

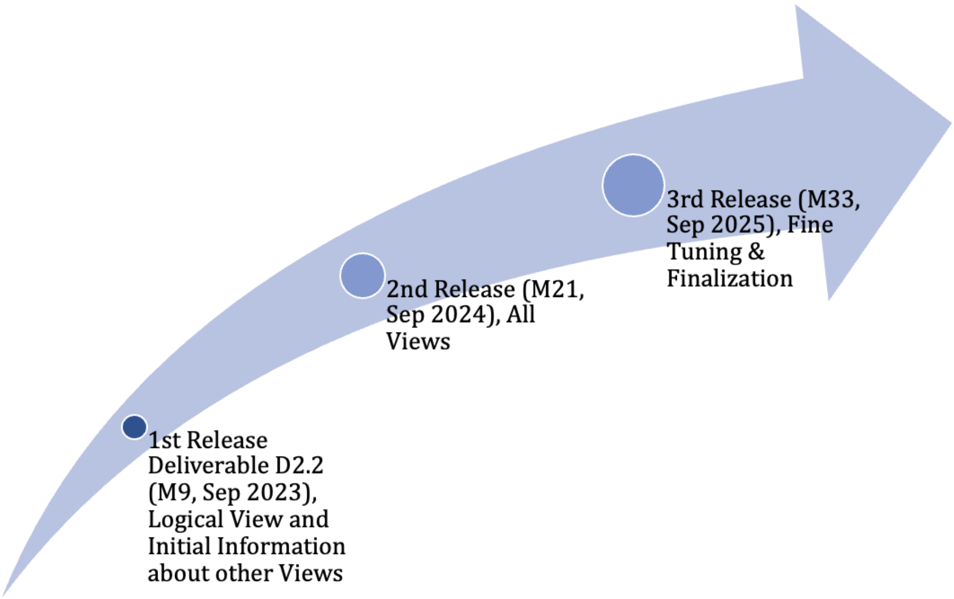

Figure 2: Architecture Development and Releases Outlook

The development of the logical view of the architecture that is presented in later sections of this deliverables has been based on two complementary approaches that are briefly discussed in this paragraph and further elaborated in subsequent sections:

- **Top-Down Refinement**: This approach started with a review of available reference models and architectures that have been proposed for the digitization of the mining industry. Considering existing architectures and their pertinence to the Mine.io requirements collected in WP2 and described (at high level) in the DoA (Description of the Action), we produced an initial logical view of the Mine.io architecture as illustrated in Figure 3. This initial view was further refined and fine-tuned based on feedback from stakeholders during collaborative co-design sessions. It was also confronted and aligned to the outcomes of the bottom-up synthesis process.
- **Bottom-Up Synthesis**: This approach started from the requirements of the pilot sites and the mini-architectures of the solutions that the pilots are designing and developing. It aimed at identifying common technological building blocks across the different pilot sites. Hence, the Mine.io Architecture ended-up with a bottom-up synthesis of a quite general architecture that could be applied across pilot sites. This architecture was confronted and aligned to the outcomes of the top-down refinement process.

  Our methodology end-up aligning and converging the outcomes of the top-down refinement and of the bottom-up synthesis as evident in the following sections as well.

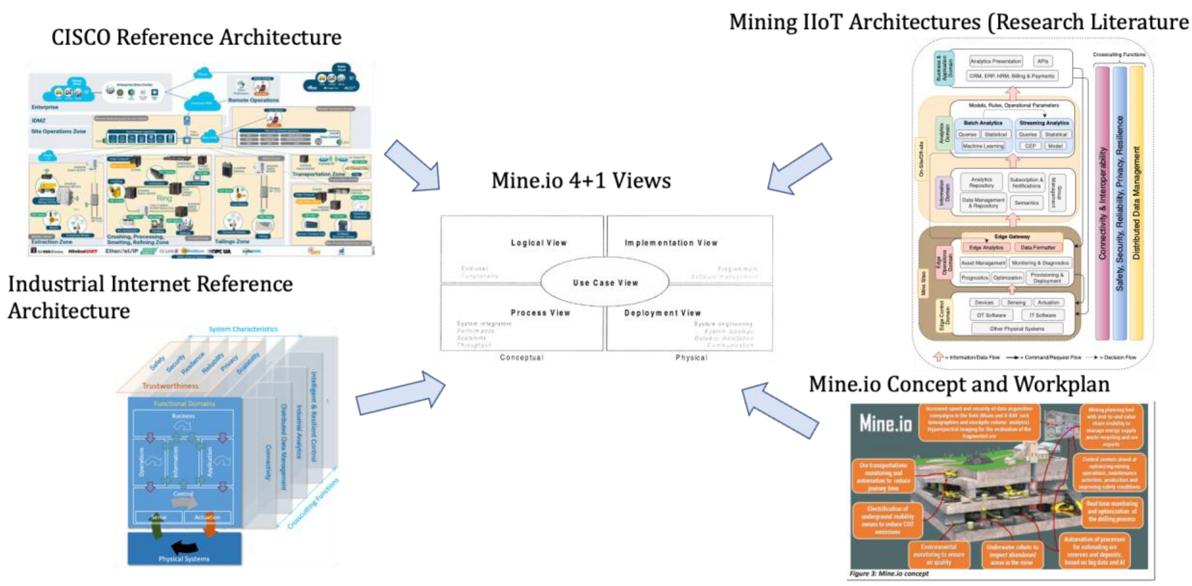

Figure 3: Top-Down Architecture Refinement

1.2.2 Co-Creation, Co-Development and Stakeholders' Engagement

The development of the Mine.io architecture has acknowledged the importance of capturing different stakeholders' viewpoints, including industrial end-users of the mining industry and providers/integrators of digital solutions for mining. To this end, the project has organized a number of co-development sessions where different views and aspects of the architecture have been presented and relevant feedback has been obtained. The following **Errore. L'origine riferimento non è stata trovata.** provides a list of such sessions, which formed a core part of the project's methodology for deliverable version 1.

Session No.	Scope of Activities	Date	Location/Mode
1	Methodology Presentation; initial Logical Architecture	May 12th, 2023	Online
2	Review of Logical Architecture and Confrontation against pilot needs	June 6th, 2023	Online
3	Review of Use Cases Logical Architectures	June 30th, 2023	Athens, Greece

Table 1Overview of Architecture Development Meetings & Sessions

1.2.3 Technology Components Specification and Description

Mine.io developed an inventory of the project partners' technological components, including the partners' background technologies that are enhanced or customized in-line with the project's needs. This inventory is included within the project's work plan, specifically as part of task T2.6. It was created and is maintained through a properly designed spreadsheet. For the purpose of providing a more technical and focused description of Mine.io technologies in this deliverable, a simple template for describing technologies has been employed, which is included in chapter 4 of this document.

1.3 RELATION TO OTHER WPS/TASKS

This deliverable documents outcomes of several tasks of WP2 of the project, notably of:

- **"T2.5 Holistic management systems, architecture, and resource re-utilization requirements"**, as it documents the Mine.io architecture and illustrates its deployment and reuse across the different pilot sites.
- **"T2.6 Inventory list of software, hardware, standards and use case data"**, as it describes several of the main components and technologies of the Mine.io project. Note that the entire inventory list of software and hardware components that are used in the project are available as part of a spreadsheet that is maintained as a living document. An indicative list of the technology components of this document is provided in Chapter 4 of this document.

The deliverable considers the outcomes of the other tasks of WP2 (i.e., T2.1-T2.4) in terms of the requirements of the pilot sites that drove the architecture development. Some of those requirements are summarized in subsequent sections of the deliverable. Most importantly, the deliverable provides valuable input to all technical work packages of the project (i.e., WP3-WP6). It is of prime importance for work packages that involve integration work (i.e., WP5 on the Mine.io platform and WP6 on the Mine.io pilots), given that the structuring principles laid down in this deliverable will be used to drive this integration work.

1.4 DELIVERABLE STRUCTURE

The deliverable is structured as follows:

- Chapter 1 provides an introduction to the project itself and the current deliverable.
- Chapter 2 following this introductory chapter presents State-of-The-Art reference models for industrial architecture, with emphasis on architectures for digital mines. Each of the architectures is presented in terms of its relevance to Mine.io. Hence, the chapter illustrates how each reference model was considered in the top-down refinement process.
- Chapter 3 presents the main requirements that have driven the specification of the Mine.io architecture in this deliverable. As already outlined, these requirements were considered as part of a bottom-up synthesis approach.
- Chapter 4 presents the top-down requirements derived from the software and hardware components utilised for the implementation of Mine.io solutions, which have also been used for the design of the Mine.io architecture.
- Chapter 5 present the current version of the Mine.io architecture based on the abstraction layered approach.
- Chapter 6 presents the technical architectures of the solutions that will be deployed at the various pilot sites, including their alignment to the overall Mine.io architecture.
- Chapter 7 sheds light on the development and deployment infrastructure of the project.
- Chapter 8 is the final and concluding chapter of the deliverable.
- References include all the publications, papers, online books and reference material which have been used for the preparation of the current deliverable.

2 Reference Architectures for Digital Mines and Related Industries

2.1 Overview – Summarized Presentation

The specification of the Mine.io architecture for digital automation and digitally enabled operations in mining sites took advantage of existing architectures (including reference architectures) that have been proposed for mining operations and the mining industry. More specifically, Mine.io benefitted from the experience of prominent research and industrial stakeholders, that have already introduced architectures, best practices and blueprints for designing, implementing and deploying digital systems for the mining industry. In this direction, the project has reviewed some of the most prominent industrial architectures blueprints with a special focus on industrial architectures for the mining industry. The aim of this review process was to identify Information Technology (IT) and Operational Technology (OT) elements, design principles and structuring principles that could be reused and adapted to the needs of the Mine.io project.

It is acknowledged that the reviewed architectures have been developed and introduced in order to address specific sets of requirements, such as requirements for building robust mining networks or high-performance digital twins. These architectures cannot be directly reused in the scope of Mine.io. Hence, instead of copying or reusing them as-is, Mine.io has opted to take advantage of some of their concepts and building blocks, which were integrated in the Mine.io reference architecture following their proper adaptation to the requirements and needs of the Mine.io stakeholders.

Table 2 provides an overview of the architectures that were reviewed and outlines their concepts that have impacted the Mine.io architecture. It also presents the type of each one of the reviewed architectures (i.e., standards, vendor driven, research driven), as well as the owner/creator of the architecture.

No.	(Reference) Architecture Name/Description	Owner	Type	Summary of Impact on Mine.io Architecture
1	Industrial Internet Reference Architecture (IIRA)	Industrial Internet Consortium (IIC)	Standard	Mine.io has adopted IIRA's concept of functional domain and cross-cutting functions
2	Reference Architecture Model Industrie 4.0 (RAMI 4.0)	Platform Industry 4.0	Standard	Mine.io uses a similar taxonomy of industrial assets and functions
3	Cisco Industrial Automation for Mining –	Cisco	Industrial	The Cisco industrial architecture influences the networking aspects

	Cisco Reference Design (CRD)			of the Mine.io architecture
4	Rockwell Architecture for Mining	Rockwell	Industrial	The Rockwell architecture has influenced the specification of the network and the devices structure of the Mine.io architecture
5	Various Research works on Industrial Architectures for Mining	Various	Research	The Mine.io architectures have considered concepts from the layering of these architectures and the creation of Digital Twins

Table 2: Overview of the (Reference) Architectures Reviewed and of their Relevance to the Mine.io RA

Following paragraphs of this Section present a brief overview of each one of the reviewed architectures and discuss how they have been considered in the specification of the Mine.io architecture.

2.2 The Industrial Internet Reference Architecture (IIRA)

2.2.1 Overview Presentation

The Industrial Internet Reference Architecture (IIRA) is a reference framework which indicates a comprehensive and standardized approach to designing, implementing, deploying and operating industrial systems based on internet connectivity and internet connected objects. It is one of the most popular blueprints for architecting robust, secure and scalable industrial systems. It is used in the manufacturing and industrial automation sectors, yet it is also appropriate for applications in other industrial sectors as well.

IIRA is a standards-based architecture i.e., it uses a host of standards to provide structuring principles and blueprints for industrial systems. For instance, it relies on ISO/IEC/IEEE 42010[ISO] as a common language and conceptual framework for the development of system architectures. As another example, it suggests the use of the Object Management Group (OMG) Data Distribution Service (DDS) [OMG] and of the Open Platform Communications Unified Architecture (OPC-UA) as standard ways for real-time communication and data exchange across different industrial systems [OPC-UA].

In terms of structuring principles for the integration of industrial systems, the IIRA promotes the decomposition of non-trivial industrial platforms into different functional domains. Specifically, the IIRA specifies six (6) functional domains. Each domain represents a specific area of functionality of industrial internet system. The domains include: (i) Asset Management, which focuses on the management and optimization of industrial assets (e.g., sensors, machines, equipment, other physical devices and various other forms of internet connected devices); (ii) Operations Technology (OT), which focuses on the integration and coordination of operational technologies like Supervisory Control And Data Acquisition (SCADA) systems and Programmable Logic Controllers (PLCs); (iii) Control Systems, which focuses on the orchestration and management of control processes within industrial internet systems, which enables functions like process control and industrial workflows management; (iv) Information Technology (IT), which is devoted to the integration of conventional IT systems (e.g., ERP systems, cloud computing infrastructures) with industrial environments; (v) Business Operations, which focus on the strategic and business-related aspects of industrial internet systems such as supply chain management aspects and (iv) Application Platform, which focuses on tools and technologies for developing custom applications, integrating third-party applications, and managing application lifecycles.

At the same time, the IIRA specifies cross-cutting functions, which represent horizontal capabilities that are applicable to multiple functional domains. Some of the most popular cross-cutting functions according to the IIRA blueprint include security (i.e., safeguarding industrial systems from security threats and vulnerabilities), interoperability (i.e., seamlessly exchanging data and information with other industrial systems), scalability (i.e., addressing growing data volumes and user basis in a cost effective manner and despite changing requirements and workflows), analytics (i.e., analysing large volumes of data to extract actionable insights) and more. **Errore. L'origine riferimento non è stata trovata.** provides an overview of the key cross-cutting functions and functional domains of the IIRA.

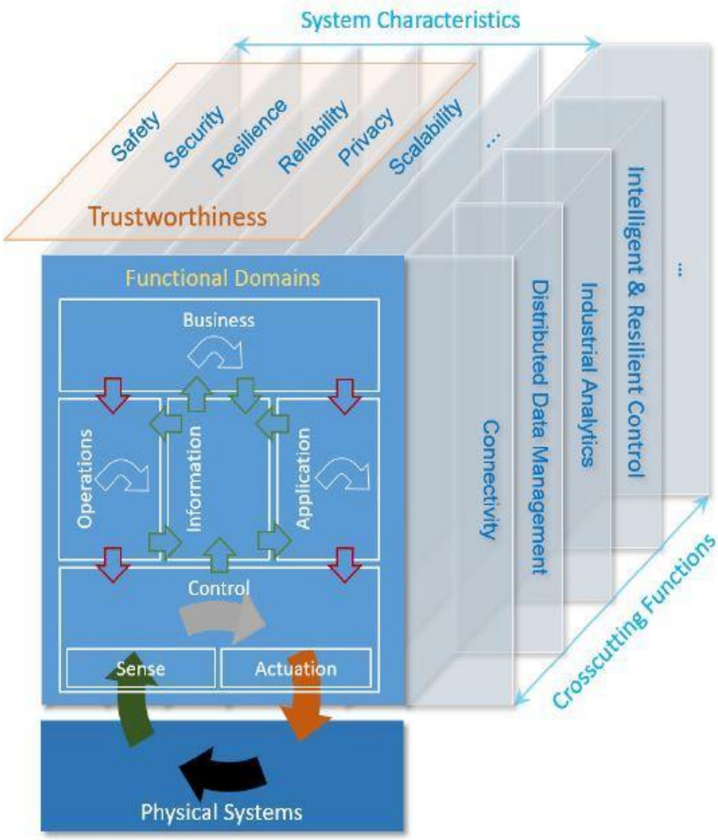

Figure 4: Functional Domain and Cross-Cutting Functions of the IIRA

2.2.2 Relevance to Mine.io

The IIRA blueprint can be used to support mining functionalities and operations, including digitally driven functions. In principle, the functional domains of the IIRA highlight functions that are relevant to the mining industry, including to mining operations are dealt within Mine.io. This is for example the case with Asset Management operations which are specified in the IIRA and implemented in the scope of Mine.io as well. Nevertheless, IIRA is designed as a general-purpose industrial automation architecture, rather than a more specific architecture for the mining industry. Hence, the Mine.io architecture will specify its own functional domains and relevant cross-cutting functions.

Overall, Mine.io has considered the concept of functional domains and cross-cutting functions in the specification of its architecture. This is the high-level conceptual contribution of the IIRA review to the logical view of the Mine.io architecture. There are also other concepts and guidelines of the IIRA that can drive the specification of other views of the Mine.io architecture, such as the use of the cloud/edge architecture for the IIRA compliance implementations. Nevertheless, such more detailed considerations are beyond the scope of this document that is mostly focused on the logical view of the Mine.io architecture.

2.3 Reference Architecture Model Industry 4.0 (RAMI 4.0)

2.3.1 Overview Presentation

The Reference Architecture Model for Industry 4.0 (RAMI4.0) is a framework destined to address the complexity of digital manufacturing systems. To this end, it provides a standards-based approach to the implementation of smart manufacturing systems, emphasizing systems that must be scalable, interoperable and flexible across different manufacturing domains. The model is best represented based on the famous cube of Figure 5. It specifies the structure of non-trivial digital manufacturing systems in a layered fashion that facilitates system integration. Some of the main layers include: (i) The Business Layer, which business processes and models that are striving to achieve operational excellence and optimal decision making; (ii) The function Layer, which comprises core functions of digital manufacturing systems like production planning and resource management; (iii) The Communication Layer, which deals with interconnectivity between devices, systems, and humans leveraging technologies like cloud/edge computing; (iv) The Information Layer, which comprises the ever important data collection, storage, and analytics functionalities. To this end, it leverages Big Data and AI (Artificial Intelligence) technologies.

There also layers that deal with individual assets and IT/OT integration as illustrated in Figure 5. Overall, RAMI 4.0 is boosting agility, efficiency, and adaptability in the implementation and integration of digital manufacturing systems. It provides a foundation for building and deploying sophisticated digital manufacturing systems like Digital Twins.

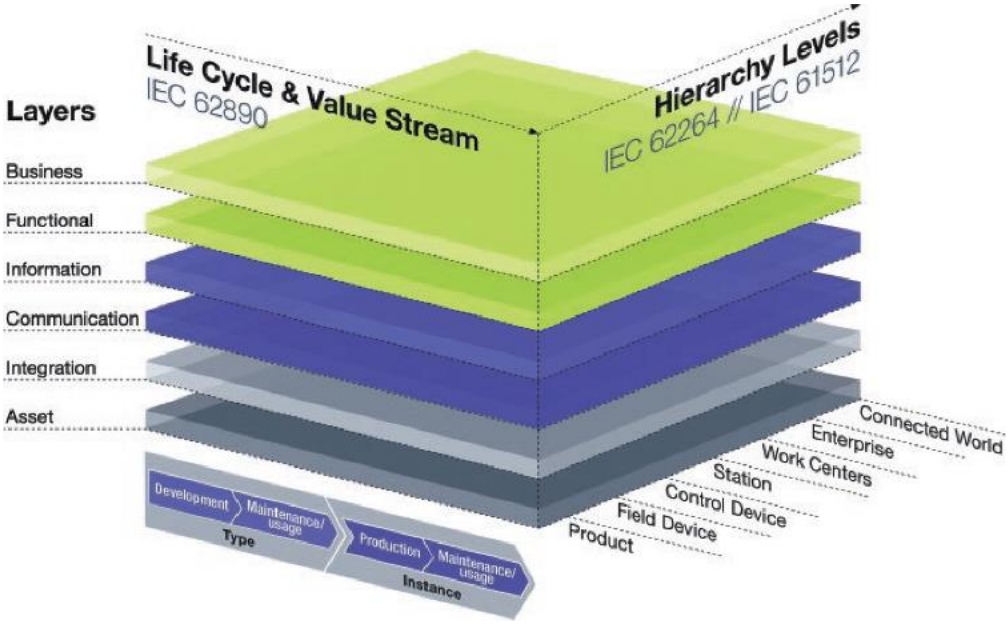

Figure 5: Overview of the Reference Architecture Model Industry 4.0 (RAMI 4.0)

2.3.2 Relevance to Mine.io

RAMI4.0 is focused on digital manufacturing and, as such, is not designed to support digital mining functionalities. However, it underlines some of the core industrial functionalities and their structuring into layers. The latter are highly relevant to digital mining operations and have therefore inspired the logical view of the Mine.io architecture. Specifically, communication, information exchange, integration and asset management aspects are at the heart of the Mine.io architecture as well.

2.4 CISCO NETWORKING ARCHITECTURE FOR MINING

2.4.1 Overview Presentation

CISCO has recently introduced its Industrial Automation for Mining in the form of a Cisco Reference Design (CRD) [Cisco20]. The latter defines an RA that supports mining processes and non-processes services in ways that ensure robust, reliable and secure communications. It is an architecture that is clearly focused on the network part of a mining site, considering the rich set of wired and wireless networking technologies. It has been designed for both open pit [Pekol19] and underground mining operations. The architecture is driven by specific use cases and provides different network designs for the mining plant operations.

Figure 6 illustrates a snapshot of the logical architecture, which presents different operational domains that map to different control areas of the mining process. The operational domains are presented from a logical perspective and include the following building blocks: (i) Remote Operations, which are destined to support remote control and visibility across pit-to-port operations from a centralized location. They are suitable for drill control, train control, and digital dispatch for the mining fleet; (ii) Enterprise Wide Area Network (WAN), which is destined to support connectivity to the Remote Operations Centre, while at the same time providing connectivity across the pit-to-port value chain; (iii) Sitewide Networking and Security Services, which comprise Enterprise WAN interface via an Industrial DMZ, Mine Core networking, and on-site Data Centre support for network and security management; (iv) Mine-wide Applications, which support the mining process. They include systems like data historians, asset Management systems, SCADA applications, and local operational centres; and (v)Various specific on-site Mining operational domains, which have specific requirements like mobility support, support for industrial automation in processes like crushing and smelting, support for transportation control systems, as well as support for safety and regulatory compliance (e.g., dust monitoring use cases).

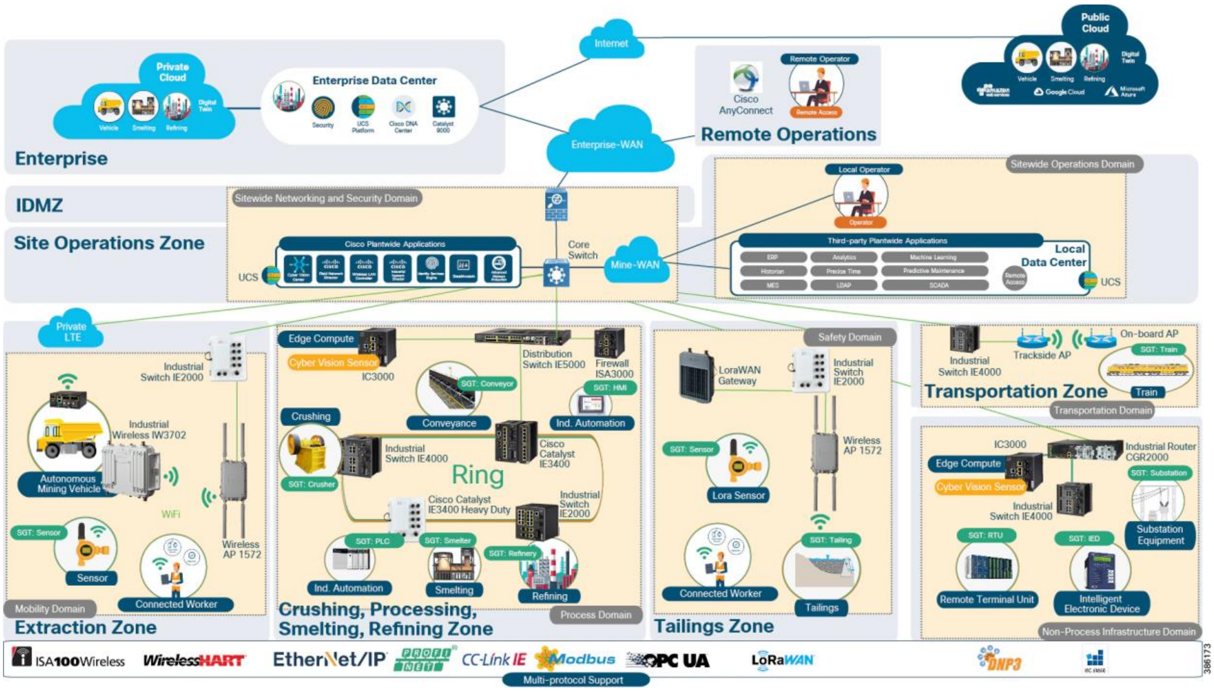

Figure 6: CISCO's Industrial Architecture for the Mining Industry

2.4.2 Relevance to Mine.io

The CISCO industrial architecture for the mining industry is mainly focused on the network and communication elements of the industrial systems. It provides insights on the different types of networking capabilities required, as well as on potential solutions. On the other hand, the Mine.io architecture does not develop its networking part on a fine level of detail. Hence, at this stage where a logical view is developed, the Cisco architecture has been consulted in the specification of the networking layer and the IoT communication protocols of the Mine.io industrial systems. The Cisco architecture will be revisited at later versions of this deliverable where the physical and deployment views of the Mine.io architecture will be presented in more detail.

2.5 Rockwell Automation Architecture for Mining Based on Converged Plantwide Ethernet

2.5.1 Overview Presentation

Rockwell [Rockwell19] has specified an industrial architecture for the convergence of OT with IT. The architecture is driven by the Mining Smart Industry Architecture (SIA) design considerations, which leverage standard Ethernet, Internet Protocol (IP), network services, security services, and EtherNet/IP™ to realize the OT/IT convergence. The latter is based on the so-called Converged Plantwide Ethernet (CPwE) reference architectures, which comprises a collection of validated designs, along with relevant blueprints and best practices.

The CPwE architectures comprise the following building blocks: Smart IIoT devices, Controllers, I/O, drives, instrumentation, actuators, analytics, and single IIoT network technology (Ethernet/IP), Zoning technologies (segmentation), connected LANs, functional areas, and security groups, firewalls and more. Rockwell Automation has collaborated with Panduit, Endress+Hauser, and Cisco in order to test and validate the converged architecture.

2.5.2 Relevance to Mine.io

The CPwE architecture bears some similarities to the Cisco reference architecture, yet it is mostly Ethernet based. Similar to the case of the Cisco CDR it is focused on the networking part of the mining system. As such it provides an alternative network design option for the networking and communications part of the Mine.io architecture. It will be considered at later stages where the physical/deployment views of the Mine.io architecture will be detailed. Moreover, it will be also considered for any pilot deployments where an Ethernet based solution can be applied.

2.6 Digital Mining Architectures from the Research Literature

2.6.1 Overview Presentation

Several researchers (including research teams of the consortium) have researched and documented industrial systems architectures for mining applications and the mining industry. For instance, in [Aziz20] a multi-layered IIoT platform architecture for the mining industry is proposed. The architecture comprises several layers, including a physical, data management, and cloud/edge layers (Figure 7). Systems at these layers are employed to simulate and facilitate interactions with mining equipment (e.g., stacker machines). Overall, the architecture is focused on the structuring of the IT elements and their interfaces to the OT systems with emphasis on data flows from the field to the applications. Moreover, the architecture specifies cross-cutting functionalities that are applicable to the various flows end-to-end. The concept of cross-cutting functions is similar to way these functions are specified and used in the IIRA.

As another example, in [Elbazi22] a digital twin architecture for mining scenarios is introduced in support of various critical scenarios involving mining equipment. The above-listed research works on mining architectures are by no means exhaustive, since several other works have dealt with specific aspects of the digital mining systems design, development and integration.

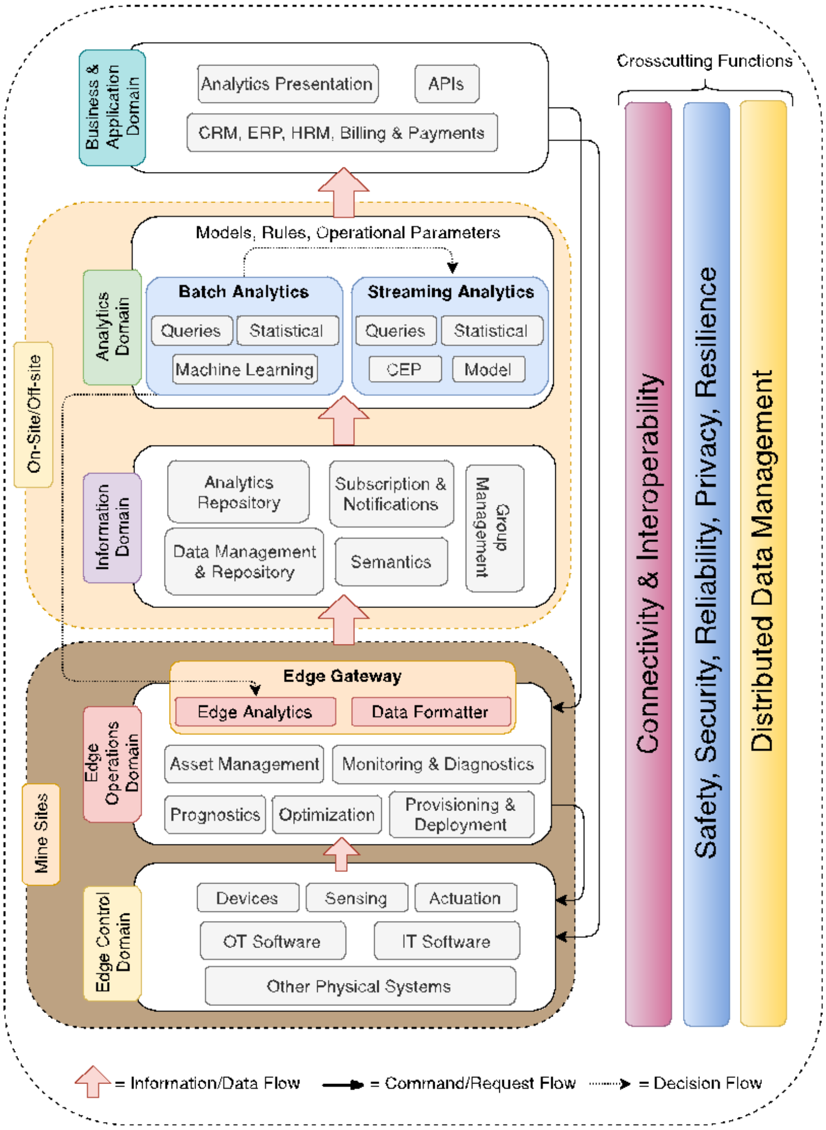

Figure 7: High-Level IIoT Architecture for the Mining Industry

2.6.2 Relevance to Mine.io

The above-listed research works are closer to the Mine.io concepts and use cases, many of which are concerned with the development of data driven systems such as digital twins. Hence, the layered structure in [Aziz20] has been strongly considered in the specification of the logical view of the Mine.io architecture, which is also largely data driven. Furthermore, the development and deployment of digital twins (as in [Elbazi22]) is considered as a core building block of the logical view of the Mine.io architecture. On the other hand, the Mine.io architecture is oriented towards mining process and mining applications rather than focused on digital components only. This is the reason why the Mine.io logical architecture is currently different from the architectures of the above-listed research papers.

2.7 Architectures from Relevant EU Projects

Additional frameworks or technologies have also been designed within relevant EU project in the smart industry domain, proposing architectures which are applicable and can be highly considered for the development of the Mine.io architecture and its required components.

In this section, Eclipse Arrowhead Architecture and FIWARE Reference Architecture for Smart Industry are analysed, while their relevance to the Mine.io project is described below.

2.7.1 Eclipse Arrowhead Architecture

Eclipse Arrowhead Framework's architecture has been designed to enable the creation of highly-scalable and secure Internet of Things (IoT) system, facilitating the seamless integration of IoT devices and services in order to ensure interoperability, reliability and cybersecurity.

Arrowhead employs a Service-Oriented Approach (SOA), where software, services and IoT devices are considered as independent components capable of intra-communication and data exchange in a decentralized way. The Arrowhead architecture can be applicable to Edge/Cloud computing, enabling real-time data exchange and efficient resource allocation through advanced networking technologies. Overall, this architecture design leads to robust systems comprising of IoT devices or IoT ecosystems enabled by software components and technology solutions providing business logic and monitoring capabilities.

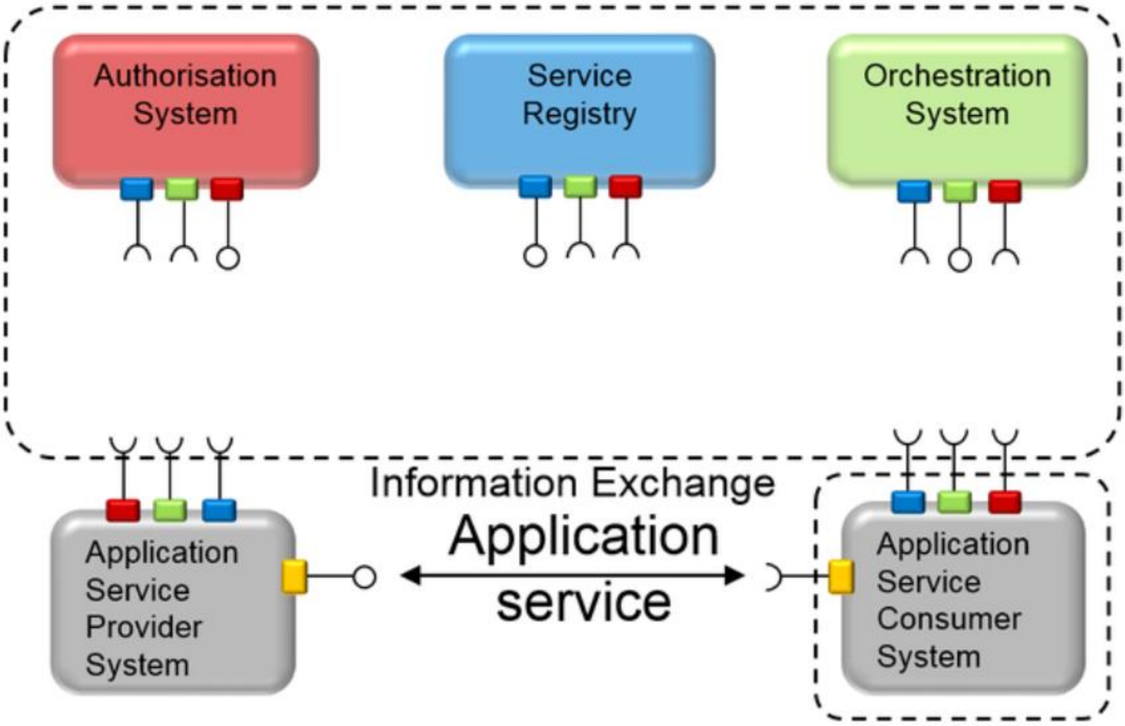

Figure 8: High-Level Architecture of the Eclipse Arrowhead Framework

As illustrated in Figure 8 above, Arrowhead architecture proposes a system design consisting of multiple IoT devices from different manufacturers and services offering different functionalities interconnected through the Arrowhead Framework, capable of seamlessly interacting or collaborating with one another. This approach facilitates service orchestration within the interconnected IoT environment, leading to easy deployment, uninterrupted updates/changes on individual components and seamless collaboration among the solutions.

2.7.2 FIWARE Reference Architecture

FIWARE technology's architecture uses a modular and scalable approach revolving around the concept of context information, which can accelerate the development of smart applications and solutions, particularly applicable in the smart cities and industrial IoT domains.

FIWARE architecture is based on Context Broker, a key component in charge of managing and orchestrating real-time context data from various sources, including IoT devices, sensors, and external systems. This offers a data-centric approach which facilitates access and utilization of real-time information by software applications, leading to highly responsive smart systems.

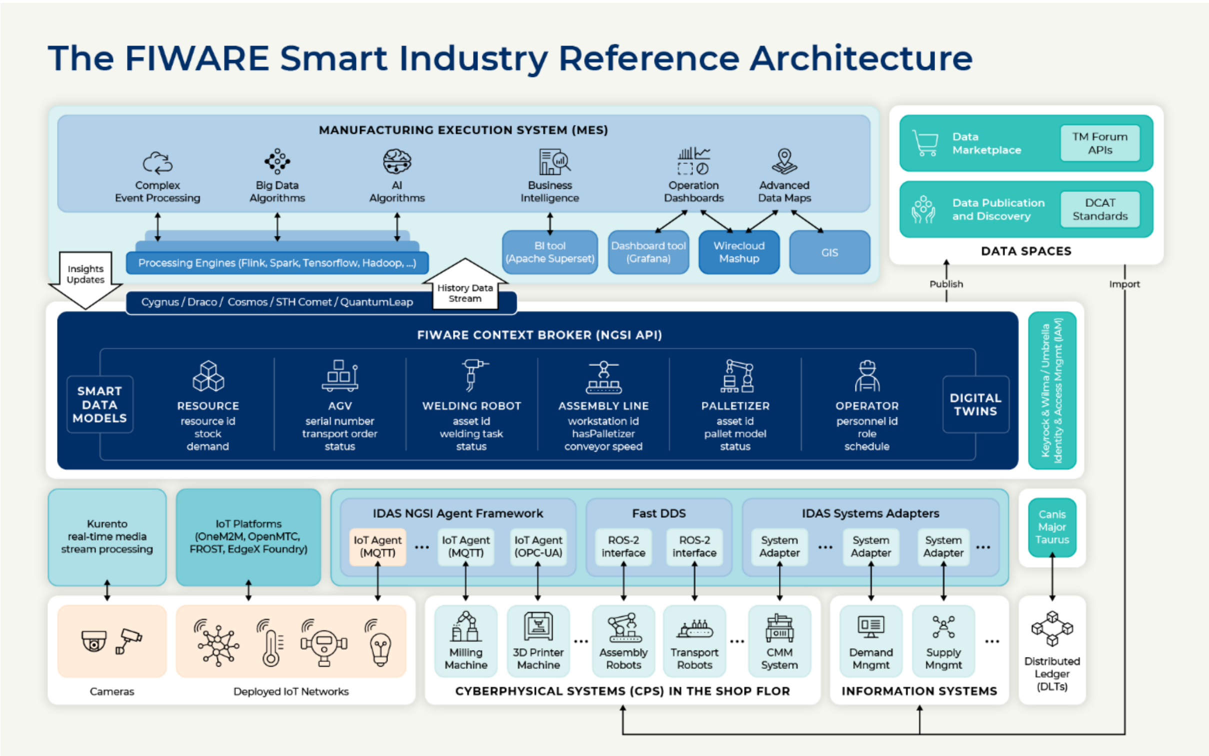

Figure 9: FIWARE Reference Architecture for Smart Industry

Figure 9 shows FIWARE architecture for smart industry, relevant to Mine.io which focuses on the smart mining domain, where it is evident that FIWARE is based on reusable software components and APIs (enablers) which provide support services of various aspects, e.g. data storage, security, IoT device management, data visualization etc. In this way, FIWARE architectural design can facilitate the development of interconnected, context-aware and interoperable applications, while facilitating open-source applications and avoiding vendor lock-in. The architecture thus, promotes real-time data exchange among devices and software, leading to seamless communication and efficiency in collaboration among the independent systems or components.

2.7.3 Relevance to Mine.io

As analyzed in the sections above, both Arrowhead and FIWARE are architectural frameworks designed to support the development/deployment of IoT and smart industry solutions, consisting of independent components which can be interconnected and capable of seamlessly exchange data for their operation.

Both Arrowhead and FIWARE have similarities with the Mine.io project, since their approaches are relevant to the Smart Industry domain. Key features of these architectural designs can be considered for Mine.io architecture, including Service-Oriented Architecture approach, high interoperability among components, system scalability to accommodate future needs within a wide range of IoT devices and applications, use of middleware components to enable IoT service management and orchestration, as well as easy integration with other technologies or platforms to enhance the capabilities of their IoT solutions.

3 ARCHITECTURE-DRIVING REQUIREMENTS

The development of the Mine.io end product will be based on the architecture presented in chapter 5 below and will aim to fulfil the different requirements presented and agreed upon in the Mine.io GA's DoA.

In order to better prepare the partners for the implementation of their respective components and solutions, the scope of this chapter is to collect, aggregate and elicit requirements from different project-related documents.

The current section, thus, analyses functional, non-functional and technical requirements which have been extracted from the project's documentation. In particular, the main available sources which have been utilised for the analysis of project requirements include the Mine.io GA, the Use Case presentations and the Mine.io project Inventory List, presented in sections 3.1 and 3.2, as well as in chapter 4 below.

The coherence of the solutions/components themselves and their respective requirements has been highlighted in each section 3.1-3.2. The link between each solution/component/Use Case, and the corresponding mining operations is highlighted in both sections.

In this way, it is shown that the Mine.io project's implementation is aligned at both implementation level (based on the solutions/components requirements presented in chapter 3) and architecture level (given the different architecture views presented in chapter 5).

3.1 BOTTOM-UP REQUIREMENTS FROM THE MINE.IO CONTRACT

The components and solutions to be developed during the Mine.io project will either be used directly in the mining operations or will carry out supplementary support activities needed to improve mining based on the Mine.io proposal.

In the sections below, we present these Mine.io solutions which have been categorised and clustered either at the relevant stages of the mining operations (sections 3.1.1-3.1.4) or at the level of support technology solutions (section 3.1.5), based on the GA's DoA.

It is noted that all solutions are represented in the Mine.io architecture, which is presented in chapter 5, proving that the development of the solutions (either clustered under mining stage operations or under support technology solutions) are coherent with the general project architecture.

3.1.1 Stage One Operation: Ore Exploration

3.1.1.1 Muon imaging and monitoring instrumentation for mining applications

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the following components: • Underground muography instrumentation.

	• Underwater muon imaging system
2.	The solution should support applicability of the developed detector.
3.	The solution should offer advanced muon imaging modules.
4.	The solution must provide 3D images of in-situ geological structures and density distributions.
5.	The solution should use novel, low-power, passive geophysical exploration, and imaging technology.
6.	The solution should consist of a device with sensitive area shorter than 0.6 square meters and a total weight less than 50kg, and external dimensions not exceeding 90cm x 90cm x 70cm.

Table 3: Bottom-Up Requirements from Solution: "Muon imaging and monitoring instrumentation for mining applications" (Stage One Operation "Ore Exploration")

3.1.1.2 Muon technology for In-Situ Underwater Exploration

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the component "Waterproof casing for an underwater muon detection system".
2.	The solution should support a new application space to combine underwater robots and waterproof muon detectors for muography in water-filled mines.
3.	The solution should provide effective methods for data transfer, including options such as underwater wireless communication, cable-based transmission, or robot-based data reading and delivery, and for cost-effective aquatic muography.
4.	The solution should support a workflow and a procedure in which sophisticated underwater robots are used to transport muon detectors underwater.

Table 4: Bottom-Up Requirements from Solution: "Muon Technology for In-Situ Underwater Exploration" (Stage One Operation "Ore Exploration")

3.1.1.3 Underwater vehicle for in-situ exploration in flooded mines

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the following components: • Autonomous exploration underwater robot. • Digital twin underwater data export software system. • Infrastructure for flooded mines exploration/ digitalization.
2.	The solution should consist of an untethered and autonomous underwater robot

	capable of exploring flooded mines.
3.	The solution must offer extension of the exploration capabilities, incorporating artificial intelligence and autonomous exploration functionalities such as 3D mine mapping, geological, mineralogical and spatial information gathering, to produce high-resolution 3D density maps of underground targets.
4.	The component must provide navigational sensors enabling semi-autonomous surveys.
5.	The solution should be able to integrate optical and acoustic communications.
6.	The solution must offer automatic data processing and integration capabilities to create a mine digital twin.

Table 5: Bottom-Up Requirements from Solution: "Underwater vehicle for in-situ exploration in flooded mines" (Stage One Operation "Ore Exploration")

3.1.2 Stage Two Operation: Ore Extraction

3.1.2.1 Enhanced drill core analysis in-situ using X-Ray technologies

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the following components: • X-ray scattering (WAXS) • X-ray Computed Tomography
2.	The X-ray scattering component should be capable to create a setup that uses a polychromatic beam and a spectrometer.
3.	The X-ray computed tomography component should offer a portable CT system.

Table 6: Bottom-Up Requirements from Solution: "Enhanced drill core analysis in-situ using X-Ray technologies - X-ray scattering (WAXS)" (Stage Two Operation "Ore Extraction")

3.1.2.2 CPS for Drill System Behavior

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the following components: • Automated big data mining and AI • Virtual test bed for demonstration of MWD results
2.	The solution should analyse raw Measurement While Drilling (MWD) data.
3.	The solution must offer advanced visualization tools in 2D/3D.

4.	The solution should be capable to estimate the rock mass characteristics.

Table 7: Bottom -Up Requirements from Solution: "CPS for Drill Behavior" (Stage Two operation "Ore Extraction")

3.1.2.3 Autonomous and Electrified Vehicle Prototype

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the following components: • Primary track pavement. • Primary electronic sub-system. • Secondary sub-system. • Autonomous driving.
2.	The solution must offer an inductive power transfer system, IPT Charge, with a primary and secondary side.

Table 8: Bottom-Up Requirements from Solution: "Autonomous and Electrified Vehicle Prototype" (Stage Two Operation "Ore Extraction")

3.1.3 Stage Three Operation: Mineral Processing

3.1.3.1 Sensor-based digital-twins for smelting

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the component "Sensor-based Digital Twins for Smelting System".
2.	The solution must support the integration of advanced sensors capabilities of real-time measurements for various metallurgical processes and being coupled to on-line process models (digital twins), using data from the advanced sensors to simulate and predict process behaviours.
3.	The component must support acoustic measurements.
4.	The component must offer frequency modulated continuous wave (FMCW) radar technology to analyse properties of molten phases.
5.	The component should provide Laser-Induced Breakdown Spectroscopy (LIBS) technology for elemental analysis of molten materials.
6.	The solution must integrate the developed sensors with suitable digitization platforms for data visualization and analysis.
7.	The solution should support the optimization of smelting operations.

Table 9: Bottom-Up Requirements from Solution: "Sensor-based Digital Twins for Smelting" (Stage Three Operation "Mineral Processing")

3.1.3.2 Smart sorting with Dual-Energy XRT

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the component "Dual energy X-ray sorting using Deep Learning".
2.	The solution must support the optimization of the dual energy X-ray data acquisition process.
3.	The solution must support deep learning models.

Table 10: Bottom-Up Requirements from Solution: "Smart Sorting with Dual-Energy XRT" (Stage Three Operation "Mineral Processing")

3.1.3.3 Artificial Intelligence technology for monitoring and control of metal ore processing

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the component "PIT System and Software".
2.	The component should offer an AI-based technology focused on monitoring metal (Cu) ore processing in the flotation system.
3.	The solution must consist of a Photonic-IT (PIT) system and software to monitor, record, process, and analyse images of the flotation froth.
4.	The component must support AI algorithms to determine the content of monitored metal in the flotation froth using image parameters.
5.	The solution should support the functionalities of X-ray analysers and visual inspection systems, enabling determination of metal content in the flotation froth.

Table 11: Bottom-Up Requirements from Solution: "Artificial Intelligence Technology for Monitoring and Control of Metal Ore Processing" (Stage Three Operation "Mineral processing")

3.1.3.4 Online model-based optimization of primary and secondary production

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution should support strategies to optimise production in mineral processing and metallurgical processes under varying ore properties.
2.	The solution must support control strategies to ensure efficiency and viability of the combined primary and secondary production.
3.	The solution must offer a framework for model-based supervisory control of mineral processing circuits, for both primary and secondary production.

4.	The solution must provide a modelling framework that considers distributions of material properties (size, mineral content, etc.).

Table 12: Bottom-UP Requirements from Solution: "Online Model-based Optimization of Primary and Secondary Production" (Stage Three Operation "Mineral Processing")

3.1.4 Stage Four Operation: Waste Management

3.1.4.1 Multi-Source Data Fusion and Interpretation for Surveillance of tailings embankments

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the following components: • Electric Resistivity Imaging (ERI). • Seismic Imaging.
2.	The solution should support a combination of an array of electrodes placed at the surface in the target area to retrieve electrical resistivity imaging sections
3.	The solution should have the capability to use passive (virtual) seismic sources and provide a map of the subsurface of infrastructures in terms of seismic velocities (body- and surface-waves) via an array of geophones around the target area, through seismic imaging.

Table 13: Bottom-Up Requirements from Solution: "Multi-Source Data Fusion and Interpretation for Surveillance of tailings embankments" (Stage Four Operation "Waste Management")

3.1.4.2 Pre-Classification of Stockpile Materials for Other Uses

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the component "Characterization system for stockpiles".
2.	The component must apply advanced data acquisition and processing technologies to gather detailed information about stockpile composition, quantity, and characteristics.
3.	The solution should utilize a lightweight active LiDAR sensor and a passive hyperspectral/multispectral sensor on a drone for aerial data collection, along with appropriate geophysical sensors for ground-level data collection.

Table 14: Bottom-Up Requirements from Solution: "Pre- Classification of Stockpile Materials for Other Uses" (Stage Four Operation "Waste Management")

3.1.4.3 UAV - Derived Electromagnetics and Magnetics for Mineral Exploration and Waste Identification

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the following components: • UAV System Prototype. • Digital architecture for the 3D slagheap model. • Simulation processes algorithms for the 3D dump model evolution.
2.	The solution should offer a digital tool for processing and displaying magnetometry and geometric data collected with drones in mining deposit environments.
3.	The solution should be capable to serve as a platform for analysis that supports risk reduction strategies combined with material recovery possibilities.
4.	The solution must support connection to and integration of multiple sensors (including RGB sensors, infrared sensors laser range finders, hyperspectral/multispectral and magnetic sensors).

Table 15: Bottom-Up Requirements from Solution: "UAV Derived Electromagnetics and Magnetics for Mineral Exploration and Waste Management Identification" (Stage Four Operation "Waste Management")

3.1.4.4 Blockchain-Based Traceability System for Sustainable and Integrated Waste Management

The solution's requirements are detailed in the table below.

ID	Requirements
1.	The solution must comprise of the component "Sustainable waste management based on distributed ledger technology".
2.	The solution must offer a blockchain-based traceability platform to enable tracking of social and environmental parameters.
3.	The solution must provide a robust Chain-of-Custody system using blockchain technology to ensure secure tracking of mining waste data throughout the supply chain.
4.	The solution should integrate secure capabilities for the traceability platform to connect with relevant databases and systems.
5.	The component should support/help data mining techniques to enable advanced information retrieval and cross-referencing.

Table 16: Bottom-Up Requirements from Solution: "Blockchain-Based Traceability System for Sustainable and Integrated Waste Management" (Stage Four Operation "Waste Management")

3.1.5 Mine.io Support Technology Solutions

On top of the solutions and components utilised for the different stages of the mining process, the Mine.io consortium will also develop components which provide additional services to the mining process, including support activities, supply chain operations, telecommunications, data fusion and analysis, big data storage and management etc.

The requirements of these Mine.io components are analysed in the tables below.

3.1.5.1 Workload scheduling and planning

ID	Requirements
1.	The component must have the capability to translate and transform end-users' requirements and acceptance conditions into technical constraints, criteria, and objective functions, and an optimization algorithm that balances conflicting criteria and constraints.
2.	The component must provide an algorithmic solution based on hybridizing and combining various AI techniques regarding a given time horizon according to the defined metrics and characteristics.
3.	The component should support allocation of resources (workers, equipment, machinery) and ensure compliance with operational and site constraints.
4.	The component should support evaluation and optimization methods for mining context, considering reference methods (genetic algorithms, tabu search) and novel approaches (Swarm Intelligence paradigm).

Table 17: Bottom-Up Requirements from Component "Workload Scheduling and Planning" (Stage Four Operation "Waste Management")

3.1.5.2 Workflow automation and smart contracts

ID	Requirements
1.	The component must offer a workflow automation engine for handling operational and business workflows in mines and raw material value chains.
2.	The component should support communication of actions to relevant stakeholders through dedicated systems and apps, integrating a blockchain-based distributed database that supports smart contracts for work orders execution and recording of collaboration and business agreements.
3.	The component should provide an engine with analytics functionalities.
4.	The component must offer a platform that automates workflows and integrates smart contracts.
5.	The component should support integrated digitized workflows across mine operation departments, decision levels, and different stakeholders, operating on shared-based information in a blockchain network.
6.	The component should provide a blockchain-based workflow engine which allows the digitalization of interactions, service agreements and contracts between

	different stakeholders.
7.	The component must support raw materials value chains to track efficiently the production, origin and exchange of materials.

Table 18: Bottom-Up Requirements from Component "Workflow automation and smart contracts" (Stage Four Operation "Waste Management")

3.1.5.3 Predictive maintenance module

ID	Requirements
1.	The component must have the capability to produce and share Life Cycle Inventory (LCI) data of project technologies through the EC Life Cycle Data Network (LCDN).
2.	The component should offer a Predictive Maintenance module for mining equipment.
3.	The component must support a Continuous Integration and Continuous Deployment (CI/CD) approach for the Predictive Maintenance module.
4.	The component should offer integration and testing plans.
5.	The component must provide real-time operational data for anticipatory and predictive capabilities.

Table 19: Bottom-Up Requirements from Component "Predictive maintenance module" (Mine.io Technology Solutions)

3.1.5.4 Social sustainability of Mine.io technological solutions

ID	Requirements
1.	The component should provide indicators for the socio-economic sustainability of Mine.io results.
2.	The component should offer metrics to quantify the social impact performance of Mine.io project.
3.	The component should seek to bring technologies to the social environment of the area of cross-cutting use for other local economic interests and the reactivation of the business activity in depopulated rural areas.

Table 20: Bottom-Up Requirements from Mine.io Component "Social sustainability of Mine.io technological solutions" (Mine.io Technology Solutions)

3.1.5.5 Remote sensing, edge computing, edge continuum/dementalized communication system

ID	Requirements
1.	The component must provide a connection platform integrating various IoT devices and data sources (remote sensing).

2.	The component should be based on a hybrid architecture, supporting configuration and customization for data collection, connectivity between the IIoT Gateway and the Cloud, and ML models.
3.	The component must offer agents in the Edge environment to manage IIoT connectivity and facilitate data flow from edge to cloud via the IIoT Gateway.
4.	The component should support data collection from sensors (remote sensing and IoT) and AI services between Edge and Cloud environments.

Table 21: Bottom-Up Requirements from Component "Remote sensing, edge computing, edge continuum/dementalized communication system" (Mine.io Technology Solutions)

3.1.5.6 Big data, data fusion, Mine.io store and supply chain

ID	Requirements
1.	The component must offer a big data cloud platform for large-scale, geographically distributed networked IIoT.
2.	The component must be capable to manage huge data storage, retrieval and administration, and facilitate data transfer via IIoT gateways.
3.	The component must provide real-time monitoring and "health" status of the mining machine, notifications about incidents retrieval.
4.	The component can be capable to integrate data received from the local selected system (from the field layer) in order to create a big data cloud for largescale geographically distributed networked IIoT. Data for integration will undergo the confidentiality requirements.

Table 22: Bottom-Up Requirements from Component "Big data, data fusion, Mine.io store and supply chain" (Mine.io Technology Solutions)

3.1.5.7 Mine.io cyber security and cyber-physical mechanisms

ID	Requirements
1.	The component must provide security mechanisms for Mine.io platform's communication channels, for both Edge and Cloud infrastructure.
2.	The component must ensure security of data exchanges within Mine.io and safeguard sensitive information from unauthorized access.

Table 23: Bottom-Up Requirements from Component "Mine.io cyber security and cyber-physical mechanisms" (Mine.io Technology Solutions)

3.1.5.8 ISO 10303 PLM/SDM Module

ID	Requirements

1.	The component should offer a data-driven approach for creating a Digital Twin (DT), utilizing open and publicly available standards for data exchange, sharing and archiving.
2.	The component should support an Open Standard Based Digital Twin as a PLM (Product Lifecycle Management) module.
3.	The component must provide an interoperability repository adhering to open and public standards, and supporting data exchange, sharing and archiving for new applications.

Table 24: Bottom-Up Requirements from Component "ISO 10303 PLM/SDM Module" (Mine.io Technology Solutions)

3.2 Bottom-Up Requirements From Use Cases

The solutions developed within the Mine.io project, the software and hardware components of which have been presented in section 3.1 in the form of their requirements, will be tested in 6 pilots comprising of relevant Use Cases, as presented in Table 25 below.

Pilot Sites	Use Cases	Offered Solutions
Pilot 1.1 Freiberg (Germany)	Use Case 1.1: Mine.io I4.0 Asset Digitalization	Digitalization of Assets and Processing Equipment
Pilot 1.2 Freiberg (Germany)	Use Case 1.2: Digital Smelter – Demonstrating an array of digitalization techniques at pilot scale with regard to smelting processes	Digitalization of Assets and Processing Equipment
Pilot 2 Polkowice (Poland)	Use Case 2: Digital Flotation System – Artificial Intelligence technology for monitoring and control of metal ore processing	Digitalization of Assets and Processing Equipment
Pilot 3 Lavrion (Greece)	Use Case 3: Geochemical Mapping of Soils and Mining Wastes in Lavrion Historic Mine Site – Post Mining Environmental Management	Digital Solutions for Waste Exploitation and Post Mining Environmental Management (remediation/rehabilitation combined strategies)
Pilot 4 Pyhäsalmi (Finland)	Use Case 4: Multi-Source Data Fusion and Interpretation for Surveillance of tailings embankments	Digital Solutions for Waste Exploitation and Post Mining Environmental Management
Pilot 5 Niederschlag (Germany)	Use Case 5: Advanced Mobility (Autonomous and Electrified Vehicles) and Operational Excellence	Sustainable Underground Mining
Pilot 6.1 Malaposta (Portugal)	Use Case 6.1: In-Situ Underwater Exploration Myon Telescope Prototype Technology Validation for Water-Filled Mines	Digital Solutions for In-Situ Mining Exploration

Pilot 6.2 Urgeirica (Portugal)	Use Case 6.2: In-Situ Underwater Exploration Robot Technology Validation for Water-Filled Mines	Digital Solutions for In-Situ Mining Exploration

Table 25: Use Cases per Pilot Site and Offered Solution

For each Use Case, the initially identified functional and non-functional requirements have been included in the current version of deliverable *D2.3 Functional and Non-Functional Requirements in the Digitisation of the Mining Sector – Version 2* and summarized in the sections below. Each Use Case presents a different scenario, where the Mine.io solutions are put into practical application to showcase their ability to enhance a different mining operation or process, and mainly to serve as a proof of the suitability and effectiveness of the relevant solutions.

In this regard, the requirements to be covered in each Use Case are directly linked to these of the relevant components and solutions, presented in section 3.1 above. Additionally, they all fit into the high-level Mine.io architecture, whose different views are presented in chapter 5 below, used as a reference for the development, integration, interconnection and operation of the entire Mine.io end product.

3.2.1 Use Case 1.1: Mine.io I4.0 Asset Digitalization

Functional and non-functional requirements for Pilot 1 Freiberg, Germany, and the Use Case 1.1, are summarized below. The table below is sourced from delivery D2.3.

Mining Phase	Functional Requirements (objectives)	Non-Functional Requirements
EXPLORATION	Be able to analyse the drill cores to identify mining potential.	Provide easy visible data
EXTRACTION	Be able to monitor the drill rig while drilling.	Provide the current data at a overall view at a dashboard which is easy visible.
	Be able to provide the current maintanance state of the drill bit to the drill operator.	Easy visible readable degredation of the drill bit
	Be able to provide the current rock (side rock or ore) material to the drill operator.	Easy interpretable data with side rock or ore information
	Be able to provide predictive maintenance capabilities by processing sensor data	(Near) real-time monitoring and statistics overview through a dashboard
	Be able to record and store data from the drill rig	Easy interpretable data with side rock or ore information
Electrification	Be able to monitor the ventilation situation in the mine.	Measurement system does not interfere with mine operations.
	Be able to provide a digital twin of the mine ventilation system with the current measurement data.	Visualise the ventilation system of the mine with ventsim.
Business Process Improvement (outside of	The platform must support the modeling, execution, and monitoring of business process workflows using BPMN 2.0 standards.	The platform must provide a user-friendly interface for designing workflows and ensure high performance and low latency in workflow execution.

industrial mining phases)	The platform must provide a de-centralized repository for storing, managing, and accessing digital permits and other regulatory documents.	The platform must ensure high availability, data redundancy, and robust access control mechanisms for document management.
	The platform must enable integration with blockchain technology for secure and transparent handling of permits, work orders, and business contracts.	The platform must ensure data integrity, security, and scalability of blockchain transactions while maintaining interoperability.
	The platform must be equipped with analytics functionalities providing the ability to track decisions and identify decision bottlenecks.	The platform must ensure high performance in data processing and analytics, provide intuitive and comprehensive visualization tools, and guarantee the accuracy and timeliness of analytics results.
Data Management	The BD platform must support discoverability, facilitating efficient retrieval and 'location' access to data stored.	The BD platform must be implemented as a location-agnostic centralized infrastructure, ensuring data access homogenization, consistency, and efficient management crosswise geographically dispersed operations.
	The BD platform must support real-time streaming, enabling immediate data processing capabilities.	The BD platform must be geographically-distributed deployed, supporting diverse location mining operations, and establishing seamless data flow.
	The BD platform must support historical data storage, managing constructively the historical data, maintaining data integrity and retrospective insights and analysis.	
	The BD platform must provide huge data storage, stressing the necessity of accommodating massive data storage, retrieval, and administration needs.	
	The BD platform must support the usage of JOTNE's PLM module as Pilot's data storage solution	
	The BD platform must provide data availability, facilitating smooth data sharing across different pilot sites and services.	

Table 26 Bottom-Up Requirements from Use Case 1.1

3.2.2 Use Case 1.2: Digital Smelter – Demonstrating an array of digitalization techniques at pilot scale with regard to smelting processes

Functional and non-functional requirements for Pilot 1.2 Freiberg, Germany, and the Use Case 1.2, are summarized below. The table below is sourced from delivery D2.3.

Mining Phase	Functional Requirements (objectives)	Non-Functional Requirements

PROCESSING	The system shall provide real-time monitoring of the bath fluid dynamics.	The system shall update and display the acoustic data online within seconds to ensure timely monitoring. It should be available during the whole metallurgical process, shall deliver reliable data and should be compatible to the Windows system.
	The system shall provide real-time data about the slag and metal phase product properties, such as bath heights and time of phase formation.	The system shall be available on-site. It shall update and display the data online within seconds to ensure timely monitoring. It should be available during the whole metallurgical process, shall deliver reliable date, and should be compatible to the Windows system.
	The system shall provide online slag (and metal phase) product properties.	The system shall be available on-site. The sensor system shall analyse the material within minutes and update and display the data within seconds for online monitoring. It should be available during the whole metallurgical process, shall deliver reliable data, and should be compatible to the Windows system.
	The system shall provide real-time data about the off-gas composition.	The system shall be available on-site. The sensor shall analyse the gas compositon and update and display the data within seconds. It should be available during the whole metallurgical process, shall deliver reliable data, and should be compatible to the Windows system.
	The system shall provide real-time data of oxygen potential measurements.	The system shall be available on-site. The sensor shall determine the oxygen partial pressure and update and display the data within seconds. A rapid replacement of the sensor technology or redundancy is desired to insure reliable data. The system should be compatibel to the Windows system.
	The system shall provide (non-contact) temperature measurement of the molten phases.	The system shall be available on-site. The sensor shall update and display the data online within seconds to ensure timely monitoring. If a continious measurement is impossible a rapid replacement of the sensor is required. The system should be compatibel to the Windows system.
Data Management	The BD platform could support discoverability, facilitating efficient retrieval and 'location' access to data stored.	The BD platform must be implemented as a location-agnostic centralized infrastructure, ensuring data access homogenization, consistency, and efficient management crosswise geographically dispersed operations.
	The BD platform must support real-time streaming , enabling immediate data processing capabilities.	The BD platform must be geographically-distributed deployed, supporting diverse location mining operations, and establishing seamless data flow.
	The BD platform could support historical data storage, managing constructively the historical data, maintaining data integrity and retrospective insights and analysis.	

Table 27 Bottom-Up Requirements from Use Case 1.2

3.2.3 Use Case 2: Digital Flotation System – Artificial Intelligence technology for monitoring and control of metal ore processing

Functional and non-functional requirements for Pilot 2 Polkowice, Poland, and the Use Case 2, are summarized below. The text is sourced from delivery D2.3.

Functional requirements from the Pilot Site to the Technology Partner include:

- PIT system will perform the flotation froth image acquisition.
- PIT system will store the collected images in the high-capacity memory.
- PIT system will make possible the Image-based prediction of metal content in flotation froth.
- PIT system will make possible the assessment of the metal content in the flotation froth samples.
- PIT system will clearly present the results.
- The results from the PIT System will be stable and within the tolerable error relative to laboratory analyses of copper content

Non-functional requirements from the Pilot Site:

- The flotation froth images and the flotation process data should be of proper quality.
- The capacity of the database is sufficient.
- A graphical user interface exists at the PIT system.

Requirements from the Technology Partner to the Pilot Site include:

- KGHM will make accessible the technological flotation line for the experiments and installation of the PIT System.
- KGHM will provide the system validation by provision of laboratory analyses for the flotation froth.

3.2.4 Use Case 3: Geochemical Mapping of Soils and Mining Wastes in Lavrion Historic Mine Site – Post Mining Environmental Management

Functional and non-functional requirements for Pilot 3 Lavrion, Greece, and the Use Case 3, are summarized below. The text is sourced from delivery D2.3.

Functional requirements refer to the main objectives of the Pilot Site, i.e., what are the features to be built to fulfil the pilot's need (e.g., assess raw material/waste/product properties). Non-functional requirements refer mainly to quality criteria that have to be fulfilled by the product, i.e., the digital twin (e.g., security, availability, accessibility, etc.). The main functional and non-functional requirements for Pilot Site 3 are presented below for each mining/metallurgical phase:

Functional Requirements

Exploration Phase:

- Detect anomalous concentrations of recoverable material.
- Create a Digital Terrain Model (DTM) of sufficient resolution.

- Investigate the geologic structure of the deposits of interest.
- 3D models for identifying areas with a potential for a more convenient and safer material restoration/recovery set-up.

Extraction Phase:

- Assess waste (chemical/mineralogical) properties.
- Risk assessment/Impact assessment/Carbon Footprint on the extraction process.

Waste Management Phase:

- Assess waste (chemical/mineralogical) properties.
- Define optimum management/remediation alternatives.

Non-functional Requirements

Exploration Phase:

- Drones with suitable characteristics for magnetic data acquisition.
- Accurate readings of the background magnetic radiation from the site".
- High-quality cameras and sensors capable of granting the precision required for subsequent modelling.
- Data acquisition line with an adequate definition/depth ratio.
- Process time lapses compatible with established work rates.
- The potential of incorporating new functionalities.

Extraction Phase & Waste Management Phase:

- Level of accuracy - Extensibility/Potential of incorporating new functionalities.
- Compliance with Environmental Legislation.
- Compliance with Health & Safety Regulations.
- Stakeholders Engagement.
- Social Compliance.

3.2.5 Use Case 4: Multi-Source Data Fusion and Interpretation for Surveillance of tailings embankments

Functional and non-functional requirements for Pilot 4 Pyhäsalmi (Oulu), Finland, and the Use Case 4, are summarized below. The text is sourced from delivery D2.3.

Functional requirements include:

- Remote data acquisition from the subsurface of tailings embankments by using electrical and seismic methods.
- The digitisation of electrical resistivity distribution in tailings embankments by using electrical resistivity electrodes as sensors.
- The digitisation of seismic velocities in tailings embankments by using geophones as sensors.

- Interpretation of geophysical signatures into elastic and hydrogeological conditions.

Non-functional requirements include:

- Pilot and demonstration program/deployment of geophysical imaging techniques, detailing the resources for deployment and task with responsibilities and timelines.
- Compliance with site requirements and permits: accessibility, fieldwork, safety due diligence (Job Safety Analysis & Safe Operating Procedures).
- Network connectivity (Ethernet) and power source availability at the testing location.
- Site dataset and information about engineering and construction drawings of the embankment, digital elevation models, and material construction specifications.
- Laboratory test protocols to develop baseline and empirical relationships. Interpretation constrained to engineering and construction parameters.

3.2.6 Use Case 5: Advanced Mobility (Autonomous and Electrified Vehicles) and Operational Excellence

Functional and non-functional requirements for Pilot 5 Niederschlag (EFS), Germany, and the Use Case 5, are summarized below. The table below is sourced from delivery D2.3.

<table>
<tr><th>Mining Phase</th><th>Functional Requirements (objectives)</th><th>Non-Functional Requirements</th></tr>
<tr><td rowspan="10">ELECTRIFICATION</td><td>The Autonomous vehicle should be able to drive with maximum speed of 15 km/h and recognize obstacles in curve and in linear driving.</td><td>Data structure of the vehicle in line with current data structure</td></tr>
<tr><td>Be able to show the capability of autonomous driving with a Demonstrator.</td><td>Enabling vehicle control</td></tr>
<tr><td>Be able to Install a track induction loop in the mine.</td><td>The induction loop should not interfere the mining process</td></tr>
<tr><td>Be able to monitor the energy transfer of the induction loop</td><td>The data should be easy visible.</td></tr>
<tr><td>Objects larger than 15cm should be identified by the system</td><td rowspan="6">Real-time processing algorithms

WI-FI communication system designated for status control must be implemented across the test track

Retrieve data from the ECU relating the vehicle status

Retrieve data from the operator

Autonomous driving module should be available from the start to the end of the vehicle trajectory</td></tr>
<tr><td>System identifies objects larger than 15cm (required)</td></tr>
<tr><td>System will identify people and machines (required)</td></tr>
<tr><td>System will identify critical obstacles smaller than 15cm (should - depending on training data for object detection and identification algorithms)</td></tr>
<tr><td>System alerts operator to human presence beyond 30m</td></tr>
<tr><td>System automatically reduces speed by 5 km/h if it detects human presence at a distance of more than 20m</td></tr>
</table>

	System automatically terminates operation if it detects human presence at a distance of less than 10m	Autonomous driving module should be continuously accessible from the operator Autonomous driving module should accurately detect obstacles (no false alarms) Autonomous driving module should have fast response time in order to avoid collisions / accidents
	System continues to operate when detecting obstacles less than 15cm and with human presence greater than 30m	
	The inductive charging system must be able to reach 90% efficiency while loading with 200 kW on a 600 V battery.	Enabling monitoring from the centralized control room of battery charging efficiency and battery status of charge.
	Be able to detect that the proper vehicle is on the track before powering the system and control that the vehicle is still available during the charging process.	Enabling monitoring from centralized control room of the truck movement through the electrificated road segments.

Table 28 Bottom-Up Requirements from Use Case 5

3.2.7 Use Case 6.1: In-Situ Underwater Exploration Myon Telescope Prototype Technology Validation for Water-Filled Mines Use Case 6.2: In-Situ Underwater Exploration Robot Technology Validation for Water-Filled Mines

The functional and non-functional requirements for Pilot 6 in Portugal, covering both Use Case 6.1 Malaposta and Use Case 6.1 Urgeirica, are summarized below. The table is sourced from delivery D2.3.

Mining Phase	Functional Requirements (objectives)	Non-Functional Requirements
EXPLORATION	Design and manufacturing an underwater muon telescope prototype	To test and characterize the underwater muon telescope prototype; Systems dimensions compatible with transporter AUV capabilities
	Design and manufacturing an underwater muon telescope prototype	To test and validate the underwater muon imaging system; Several deployments in different positions are required for the muon tomography process
	To test and validate the robotic deployment of the muon telescope	Accurate positioning and orientation of the muon telescope capabilities Launch and recover system to deploy the EVA AUV with muon telescope attached
	To test and validate the wireless operation of the Autonomous exploration underwater robot	Reliability and robustness of the wireless exploration
	To test and validate new autonomous operation of the Autonomous exploration underwater robot	Reliability and robustness of the developed autonomous exploration manoeuvres
	Test and validated the underwater data export system	Follow MINE.IO Data Standards

Data Management	The BD platform must support discoverability, facilitating efficient retrieval and 'location' access to data stored.	The BD platform must be implemented as a location-agnostic centralized infrastructure, ensuring data access homogenization, consistency, and efficient management crosswise geographically dispersed operations.
	The BD platform must support real-time streaming, enabling immediate data processing capabilities.	The BD platform must be geographically distributed deployed, supporting diverse location mining operations, and establishing seamless data flow.
	The BD platform must support historical data storage, managing constructively the historical data, maintaining data integrity and retrospective insights and analysis.	
	The BD platform must offer data type versatility, acknowledging the heterogeneity of the data generated, supporting diverse data formats and types, such as sensorial data, geospatial data, and images.	
	The BD platform must support the Pilot's need for storing the data produced (this is under discussion).	
	The BD platform must provide data availability, facilitating smooth data sharing across different pilot sites and services.	
	The BD platform must provide huge data storage, stressing the necessity of accommodating massive data storage, retrieval, and administration needs.	

Table 29 Bottom-Up Requirements from Use Case 6.1 and 6.2

4 TOP-DOWN REQUIREMENTS FROM THE INVENTORY OF COMPONENTS

Towards the development the components and solutions offered by the Mine.io consortium, which are presented in section 3.1 above, each technical partner is making use of various software, libraries, frameworks or processes. These software components are presented below per partner, while their technical requirements, specifications or restrictions are also analysed.

Additionally, many of these software components must be connected to external hardware in order to receive data, monitor systems or exchange information with other solutions of the Mine.io ecosystem. These hardware components (including IoT devices, measurement equipment, hardware devices, servers, simulation platforms etc.) are also presented below the tables of the relevant software components per partner.

In each table below, we present how each software component is related to the requirements of the solutions offered by Mine.io (analysed in section 3.1) and the requirements of the Use Cases (analysed in section 3.2) where these solutions will be practically applied. In this way, a direct correspondence is created between each software component and a unique solution or

Mine.io component, which are linked to the mining stage operations presented in sections 5 below.

It is noted that these software components presented below have been collected in the form of an Inventory List by the project partners and will form the basis for the implementation of the entire Mine.io project solutions and end results.

1. **GFT ITALIA SRL (GFT)**

GFT is utilising the following software components, the technical functionalities of which are shown in the tables below:

- Smart Contract Portal
- OpenID SSI Auth service
- Smart Contract Manager
- Tokenizer

SW Component	**Smart Contract Portal**
Reference to 3.1	(none)
Reference to 3.2	(none)
Requirements	**Description:** Backoffice application to create and manage templates and instances of Smart Contracts
	Operating System: Linux
	Input: HTML template, Solidity template, parameters values
	Output: Deployed smart contracts
	Partner/owner: GFT
	Software type: Web application
	Additional: It is designed to work with any programming language and blockchain

Table 30: Top-Down Requirements for SW Component "Smart Contract Portal" (GFT)

SW Component	**OpenID SSI Auth service**
Reference to 3.1	(none)
Reference to 3.2	(none)
Requirements	**Description:** Java REST APIs to manage an oAuth2/OIDC authentication and authorization based on W3C verifiable credentials
	Operating System: Linux
	API: Yes
	Input: client id, requested vc in the scope
	Output: access token, id token
	Partner/owner: GFT
	Software Type: REST Microservice
	Additional: Based on Veramo technology and Ethr DID method

Table 31: Top-Down Requirements for SW Component "OpenID SSI Auth service" (GFT)

SW Component	**Smart Contract Manager**
Reference to 3.1	(none)
Reference to 3.2	(none)

Requirements	**Description:** Java REST APIs to create and manage templates and instances of Smart Contracts and deploy them in a Blockchain
	Operating System: Linux
	API: Yes
	Partner/owner: GFT
	Software Type: REST Microservice
	Additional: It is designed to work with any programming language and blockchain but currently it works only for Solidity and Ethereum-based blockchains (for build and deploy)

Table 32: Top-Down Requirements for SW Component "Snart Contract Manager" (GFT)

SW Component	**Tokenizer**
Reference to 3.1	(none)
Reference to 3.2	(none)
Requirements	**Description:** Java REST APIs to create and exchange ERC20 tokens, "labelled" with the creator address
	Operating System: Linux
	API: Yes
	Partner/owner: GFT
	Software Type: REST Microservice
	Additional: It is based on a customized ERC1155 smart contract and could be improved to create and manage NFT

Table 33: Top-Down Requirements for SW Component "Tokenizer" (GFT)

2. **Technische Universität Bergakademie Freiberg (TUB)**

TUB is utilising the following software components, the technical functionalities of which are shown in the tables below:

- DECOMDA Storage Cloud Solution
- Visualisation dashboard of drill rig operation
- ACT - Advanced Process Control
- HSC Simulation
- OPC UA Server Software - Open Platform Communications - Unified Architecture
- ALMEMO Win Control
- Siemens STEP 7 V19 (TIA Portal)

Utilised hardware components:

- DECOMDA: measurement unit and data logger
- DECOMDA: collect the data of the measurement units
- Air flow sensor
- Ventsim: simulation
- Sandvik DE 110: portable drill for 46 mm drilling
- Linear potentiometer, hydraulic pressure, acceleration, and hydraulic flow sensor, Bronkhorst
- ALMEMO Data Logger 5690-2, 5590

- ABB AO2020 - Uras26: Gas sensors
- Siemens S7 1500
- Experimental LIPS-sensor
- FMCW-sensor
- Haereus ElectroNite CelOx

SW Component	**DECOMDA Storage Cloud Solution**
Reference to 3.1	(none)
Reference to 3.2	Section 3.2.1 Use Case 1.1: Mine.io I4.0 Asset Digitalization
Requirements	**Description:** storage cloud
	Cloud API: for data storage specified system
	Cloud capabilities: server
	Operating systems: FreeBSD
	Available storage space: 1TB
	Available RAM: 16 GB
	Available Processing Cores/Threads: 6 cores

Table 34: Top-Down Requirements for SW Component "DECOMDA" (TUB)

SW Component	**Visualisation Dashboard of Drill Rig Operation**
Reference to 3.1	Section 3.1.2.2 CPS for Drill System Behavior
Reference to 3.2	Section 3.2.1 Use Case 1.1: Mine.io I4.0 Asset Digitalization
Requirements	**Description:** Dashboard for life data and cloud storage
	Cloud API: for data storage specified system
	Operating systems: Linux

Table 35: Top-Down Requirements for SW Component "Visualisation Dashboard of Drill Rig Operation" (TUB)

SW Component	**ACT - Advanced Process Control**
Reference to 3.1	Section 3.1.3.1 Component: Sensor based-digital-twins-for-smelting system
Reference to 3.2	Section 3.2.2 Use Case 1.2: Digital Smelter – Demonstrating an array of digitalization techniques at pilot scale with regard to smelting processes
Requirements	**Description:** Managing and pre-processing incoming data, control routines and communication to simulation software, generate live overview of sensory data and control user interfaces
	Operating System: Windows
	Input: sensor data via OPC-UA-Server as OPC-UA-Client
	Output: data for regulators and simulation, control data
	Additional: main software for controlling and data management

Table 36: Top-Down Requirements for SW Component "ACT – Advanced Control Process" (TUB)

SW Component	**HSC Simulation**
Reference to 3.1	Section 3.1.3.1 Component: Sensor based-digital-twins-for-smelting system

Reference to 3.2	Section 3.2.2 Use Case 1.2: Digital Smelter – Demonstrating an array of digitalization techniques at pilot scale with regard to smelting processes
Requirements	**Description:** generate graphic user interfaces, digital twin software, simulation base
	Operating System: Windows
	Input: sensor and control data from ACT
	Output: control data, digital twin simulation data, process predictions

Table 37: Top-Down Requirements for SW Component "HSC Simulation" (TUB)

SW Component	**OPC UA Server Software – Open Platform Communications - Unified Architecture**
Reference to 3.1	Section 3.1.3.1 Component: Sensor based-digital-twins-for-smelting system
Reference to 3.2	Section 3.2.2 Use Case 1.2: Digital Smelter – Demonstrating an array of digitalization techniques at pilot scale with regard to smelting processes
Requirements	**Description:** managing standardised process and control data, platform independent service-oriented architecture
	Operating System: Windows/Linux/Apple/Android
	Real Time: Yes
	Open Source: Yes
	API: C, C++, .NET, (Java, JavaScript, Python), Siemens Simatic
	Transport Protocol: TCP/IP
	Input: sensor and control data
	Output: data for regulators and logging to MySQL, MongoDB, etc.
	Standardisation: OPC Foundation
	Additional: Client/Server principle for dynamic communication

Table 38: Top-Down Requirements for SW Component "OPC UA Server Software - Open Platform Communications - Unified Architecture" (TUB)

SW Component	**ALMEMO Win Control**
Reference to 3.1	Section 3.1.3.1 Component: Sensor based-digital-twins-for-smelting system
Reference to 3.2	Section 3.2.2 Use Case 1.2: Digital Smelter – Demonstrating an array of digitalization techniques at pilot scale with regard to smelting processes
Requirements	**Description:** receiving sensor data and translating it to OPC for ATC
	Operating System: Windows
	Input: sensor data
	Output: sensor data via OPC-UA server, control data via analogue

Table 39: Top-Down Requirements for SW Component "ALMEMO Win Control" (TUB)

SW Component	**Siemens STEP 7 V19 (TIA Portal)**
Reference to 3.1	Section 3.1.3.1 Component: Sensor based-digital-twins-for-smelting system
Reference to 3.2	Section 3.2.2

	Use Case 1.2: Digital Smelter – Demonstrating an array of digitalization techniques at pilot scale with regard to smelting processes
Requirements	**Description:** programming software for Siemens S-1200
	Operating System: Windows
	Input: analogue and digital sensor data
	Output: sensor data and control data to OPC UA Server Software

Table 40 Top-Down Requirements for SW Component "Siemens STEP 7 V19 (TIA Portal)" (TUB)

3. ACCELIGENCE LtD (ACC)

ACC is utilising the following software components, the technical functionalities of which are shown in the table below:

- ROS2/ROS frameworks
- ANIM (Autonomous Navigation Inside a Mine)

Utilised hardware components:

- FLIR Duo Pro R: RGB, thermal, night vision camera
- NVIDIA Jetson AGX Xavier: enable the capability to host machine learning-based algorithms for computer vision that require high computational capacity

SW Component	ROS2/ROS frameworks
Reference to 3.1	Section 3.1.2.3 Software component: Autonomous driving Section 3.1.4.3 UAV system prototype
Reference to 3.2	Section 3.2.4 Use Case 3: Geochemical Mapping of Soils and Mining Wastes in Lavrion Historic Mine Site – Post Mining Environmental Management
Requirements	**Key Principles:** uORB middleware
	Real-time: YES
	Distribution: NO
	Open Source: YES
	Message Patterns: JSON
	Transport Protocols: UART, UDP
	3rd party and Legacy systems adaptability: YES
	Standardization: POSIX interface, UAVCAN standard

Table 41: Top-Down Requirements for SW Component "ROS2/ROS frameworks" (ACC)

SW Component	ANIM (Autonomous Navigation Inside a Mine)
Reference to 3.1	Section 3.1.2.3 Software component: Autonomous driving
Reference to 3.2	Section 3.2.6 Use Case 5: Advanced Mobility (Autonomous and Electrified Vehicles) and Operational Excellence

Requirements	**Description:** ANIM component is assigned with the task of extracting a safe path and computing the proper control inputs for the autonomous navigation of a truck inside a mine, based on the measurements received from the various sensory inputs.
	Real-time: YES
	Input: • From IPT: Start / Stop messages • From sensors: Distances (from sensors), point-cloud (from depth-camera)
	Output: • To IPT: Vehicle control input (e.g. steering) • To High-level control: Vehicle state (distances from wall / rail), path to follow
	Implementation Technology: • Programming languages: C++, Python • Platform: Linux on Jetson / Raspberry • Libraries: Keras

Table 42: Top-Down Requirements for SW Component "ANIM" (ACC)

4. Politecnico di Torino (POL)

POL is utilising the following software component, the technical functionalities of which are shown in the table below:

- openLCA

SW Component	openLCA
Reference to 3.1	Section 3.1.5.4 Social sustainability of Mine.io technological solutions
Reference to 3.2	Section 3.2.4 Use Case 3: Geochemical Mapping of Soils and Mining Wastes in Lavrion Historic Mine Site – Post Mining Environmental Management
Requirements	**Description:** software developed by GreenDelta, for the evaluation of environmental and social impacts
	Operating System: Windows
	Input: Database with extension.zolca
	Output: .zolca or excel file

Table 43: Top-Down Requirements for SW Component "openLCA" (POL)

5. Institute of Communication and Computer Systems (ICCS)

ICCS is utilising the following software component, the technical functionalities of which are shown in the table below:

- Predictive Maintenance module

Utilised hardware components:

- Test data for Metal concentrations, metal compounds: Previous Assessments on Pilot

- Test data for Risk Values/Probabilities: Implementation of deterministic Risk Assessment Models, based on actual pollutants/metals concentration
- TSI model 8520 DustTrak Aerosol Monitor: Measures mass concentrations of dust, smoke, mist and fumes in real time and stores the information in a built-in datalogger. The DustTrak measures particle concentration corresponding to PM10/PM2.5, PM1.0 or respirable size fractions

SW Component	**Predictive Maintenance module**
Reference to 3.1	Section 3.1.5.3 Predictive maintenance, net zero carbon and continues development
Reference to 3.2	Section 3.2.1 Use case 1: Mine.io I4.0 Asset Digitalization
Requirements	**Description:** The Predictive Maintenance module is implemented in the form of an analytics process development tool employing AutoML and deep learning. It covers the whole lifecycle of data analytics enabling the development of several independent analytics process with different scopes (descriptive, predictive, prescriptive) from multiple data sources and algorithms. The Predictive Maintenance module for mining equipment detects anomalies, predicts the future health state, and provides recommendations about maintenance actions. To do this, it embeds (deep) machine learning algorithms for predictive maintenance in the mining industry.
	Operating System: Linux
	API: GraphQL API
	Input: Sensor data
	Output: Descriptive, predictive and prescriptive analytics outcomes
	CI/CD tools: pip, GitHub
	RDBS: MongoDB
	Exposed Services: Analytics Management Queries: consists of CRUD queries, providing the ability to create, read, update and delete the various entities in the database Analytics Process Queries: responsible for realizing the requests of the steps of the analytics process, i.e., building and triggering an analytics process and requesting analytics results

Table 44: Top-Down Requirements for SW Component "Predictive Maintenance module" (ICCS)

6. **Fundacion Tecnalia Research & Innovation (TEC)**

TEC is utilising the following software components, the technical functionalities of which are shown in the tables below:

- Traceblock
- Task Planification Tool

SW Component	**Traceblock**
Reference to 3.1	Section 3.1.4.4 Blockchain-Based Traceability System for Sustainable and Integrated Waste Management
Reference to 3.2	Section 3.2.6 Use case 5: Advanced Mobility (Autonomous and Electrified Vehicles) and Operational Excellence)
Requirements	**Description:** Blockchain based Traceability Tool
	Operating System: It will be provided as a service
	Input: Exploration, extraction, and processing to waste management and post mining activity (the specific data models will be defined by TECNALIA)
	Output: The same as input but guaranteeing its integrity and traceability thanks to Blockchain features (the specific data models will be defined) --> API queries or graphical access will probably be provided

Table 45: Top-Down Requirements for SW Component "Traceblock" (TEC)

SW Component	**Task Planification Tool**
Reference to 3.1	Section 3.1.5.1 Workload scheduling and planning
Reference to 3.2	Section 3.2.6 Use case 5: Advanced Mobility (Autonomous and Electrified Vehicles) and Operational Excellence)
Requirements	**Description:** Tool to produce a planification for the Task to be performed within a mine. Performing the optimum assignment to each operator and adequate machinery. Algorithms for mining activities scheduling. This set of methods and algorithms should help to: - Schedule mining activities, optimizing stock, team performance and project management indicators while ensures safety and union constraints. - Route equipment, material and staff by using site transportation facilities as internal trains or trucks. - Improve operational planning, forecasting, monitoring, and controlling, in order to raise the quality and reliability of decision-making in construction projects.
	Operating System: Ubuntu
	API: REST API
	Input: Operation schedules
	Output: JSON
	CI/CD tools: Git (GitHub), Jenkins, Maven, Pip, Npm, Gerrit
	Programming language and IDEs:

	Java for the core of the optimization engine the use of libraries as optaplanner will be considered HTML + Javascript for the UI
	RDBS: Mysql
	Additional: Docker containers Docker-compose: as orchestrator OSRM router (C++ based) for computing optimal trajectories/displacements within the mine.

Table 46: Top-Down Requirements for SW Component "Task Planification Tool" (TEC)

7. University of Oulu (UOULU)

UOULU is utilising the following software component, the technical functionalities of which are shown in the table below:

- GSPy
- ElectrePro / ProSysIII / ResiPy
- Geopsy / ObsPy
- QGIS / IDV / Paraview

Utilised hardware components:

- Syscal Pro (Electric Resistivity Imaging Unit)
- Geospace Seismic Recorder with Battery (seismic recorder)
- Broadband Guralp 3ESPC seismometers and Geophones

SW Component	**ElectrePro / ProSysIII / ResiPy (SimPEG)**
Reference to 3.1	Section 3.1.4.1 Multi-Source Data Fusion and Interpretation for Surveillance of tailings embankments
Reference to 3.2	Section 3.2.5 Use Case 4: Multi-Source Data Fusion and Interpretation for Surveillance of tailings embankments
Requirements	**Description:** 2D, 3D inversion software for ERI data
	Operating System: Windows

Table 47: Top-Down Requirements for SW Component "Res2DInv / ResiPy" (UOULU)

SW Component	**Geopsy / ObsPy**
Reference to 3.1	Section 3.1.4.1 Multi-Source Data Fusion and Interpretation for Surveillance of tailings embankments
Reference to 3.2	Section 3.2.5 Use Case 4: Multi-Source Data Fusion and Interpretation for Surveillance of tailings embankments

Requirements	**Description:** Seismic refraction modelling and interpretation software packages
	Operating System: Windows

Table 48: Top-Down Requirements for SW Component "Geopsy / SeisImager" (UOULU)

SW Component	**GSPy**
Reference to 3.1	Section 3.1.4.1 Multi-Source Data Fusion and Interpretation for Surveillance of tailings embankments
Reference to 3.2	Section 3.2.5 Use Case 4: Multi-Source Data Fusion and Interpretation for Surveillance of tailings embankments
Requirements	**Description:** Framework for standardisation of geophysical datasets. Organising, viewing, and processing geophysical data
	Operating System: Windows

Table 49: Top-Down Requirements for SW Component "GSPy" (UOULU)

SW Component	**QGIS / IDV / Paraview**
Reference to 3.1	Section 3.1.4.1 Multi-Source Data Fusion and Interpretation for Surveillance of tailings embankments
Reference to 3.2	Section 3.2.5 Use Case 4: Multi-Source Data Fusion and Interpretation for Surveillance of tailings embankments
Requirements	**Description:** Applications for geospatial data analysis and visualisation
	Operating System: Windows

Table 50: Top-Down Requirements for SW Component "QGIS / IDV / Paraview" (UOULU)

8. Myon Solutions Oy (MUO)

MUO is utilising the following software component, the technical functionalities of which are shown in the table below:

- CORSIKA

Utilised hardware component:

- Muon telescope: The telescope consists of layers of detectors that measure the arrival times and trajectories of muons
- Muon Tomograph Large: Gaseous particle tracking detector, which measures individual muons penetrating down to the detector position. Measures density distribution at different angles with better than 1 degree resolution
- Muography instrument for underwater imaging: Particle tracking detector for measuring cosmic muons, inside an appropriate water-tight container
- MTL cosmic muon detector: Particle tracking detector for underground density mapping

SW Component	**CORSIKA**

Reference to 3.1	Section 3.1.1.2 Muon technology for In-Situ Underwater Exploration
Reference to 3.2	Section 3.2.8 Use Case 6.2: In-Situ Underwater Exploration Robot Technology Validation for Water-Filled Mines
Requirements	**Description:** Used to simulate the production and propagation of muons through various materials, such as rocks. The simulated data can then be used to optimize the design of muon telescopes and to develop algorithms for reconstructing images of the interior of objects based on the trajectories of the muons that have passed through them.
	Operating System: CORSIKA is a cross-platform software package that can be run on a variety of operating systems, including Linux, macOS, and Windows. The software is distributed as source code, which can be compiled on the user's system using a standard C++ compiler. Alternatively, precompiled binaries are available for some popular platforms, such as Ubuntu Linux and macOS. The software is often used in high-performance computing environments, such as clusters or supercomputers, to simulate large numbers of cosmic ray events, which require significant computational resources.
	Input: It needs: (1) CORSIKA needs information about the properties of the primary cosmic ray that is responsible for producing the muons that will be detected. (2) A detailed model of the Earth's atmosphere. (3) Information about the geometry and properties of the muon detector being used, including the size, shape, and orientation of the detector elements. (4) The properties of the materials that the muons will be passing through, including the density, composition, and thickness of the materials
	Additional: Widely used in muography and other fields of astroparticle physics to simulate the interactions of high-energy particles, such as cosmic rays, with the Earth's atmosphere and with detectors

Table 51: Top-Down Requirements for SW Component "CORSIKA" (MUO)

9. Lulea University of Technology (LTU)

LTU is utilising the following software component, the technical functionalities of which are shown in the table below:

- Eclipse Arrowhead

SW Component	Eclipse Arrowhead
Reference to 3.1	Section 3.1.2.2 CPS for Drill System Behavior Section 3.1.3.4 Online model-based optimization of primary and secondary production
Reference to 3.2	Section 3.2.1 Use case 1.1: Mine.io I4.0 Asset Digitalization

Requirements	**Description:** Automation integration framework enabling large scale automation systems including legacy and IoT technology	
	Input: Engineering data	
	Output: Working automation	

Table 52: Top-Down Requirements for SW Component "Eclipse Arrowhead" (LTU)

10. Jotne Connect (JOT)

JOT is utilising the following software component, the technical functionalities of which are shown in the table below:

- EDMtruePLM (ISO 10303 repository)

SW Component	**EDMtruePLM (ISO 10303 repository)**
Reference to 3.1	Section 3.1.5.8 Establishment and maintenance of an ISO 10303 PLM/SDM Module as the interoperability infrastructure to improve data exchange, sharing and archiving processes
Reference to 3.2	(none)
Requirements	**Description:** Open Standards Based ISO 10303 repository that can manage CAD, CAE, PLM and sensor data as well as maintenance and support data
	Operating System: Windows/Linux
	Input: All relevant engineering and sensor data
	Output: All relevant engineering and sensor data
	Input Formatting: STEP AP239, XML, JSON using ISO 10303
	Input Interfaces: File based (ISO 10303-21, ISO 10303-28), C++, .NET, SOAP web-services, REST API, Arrowhead Framework
	Output Formatting: STEP AP239, XML, JSON using ISO 10303
	Output Interfaces: File based (ISO 10303-21, ISO 10303-28), C++, .NET, SOAP web-services, REST API, Arrowhead Framework
	Hardware Dependencies: Server License

Table 53: Top-Down Requirements for SW Component "EDMtruePLM (ISO 10303 repository)" (JOT)

11. Innov-Acts Limited (INN)

INN is utilising the following software components, the technical functionalities of which are shown in the tables below:

- Sensor-Data Gathering Software
- Elasticsearch

Utilised hardware components:

- Wirnet iStation: LoRaWAN gateway that specifically meets the standard requirements of public operators, MVNOs, cable operators, private businesses or public authorities that want to boost their businesses with IoT and to deploy evolutive and reliable networks, connect high volumes of end-devices, and manage millions of bidirectional messages every day

- Plug&Sense 4-20mA: The current loop is one of the most robust sensor-signalling standards. Current loops are ideal for data transmission because of their inherent insensitivity to electrical noise
- Y1400 LoRaWAN converter 0-10V, 4-20 mA: connect external analogue probes to your LoRaWAN network. Fit for industrial environments, built to have many years of maintenance free battery life
- Gas sensor BS-390, BS-391: three-wire detectors 4-20mA suitable to detect combustible gases (liquefied gas or natural gas) using a catalytic sensor set to limit up to 20% LEL for use in central alarm systems in industrial places
- Temperature & Humidity EE100Ex sensor: intrinsically safe sensor reliably measures relative humidity (RH) and temperature (T) in explosion-hazard areas. It complies with the ATEX (Europe) and IECEx (international) classifications for applications in gas up to Zone 1
- Differential pressure gause DPGT43.100: The electronic WIKA sensor, integrated into the high-quality mechanical differential pressure gauge, combines the advantages of electrical signal transmission with a local mechanical display that remains readable during a power failure. An additional measuring point for mechanical pressure display can thus be saved
- Ultrasonic Sensoneo Single Sensor 3.0: enterprise class device monitoring fill-level of various types of containers. It uses single ultrasonic beam with measurement from 1 cm up to the depth of 255 cm

SW Component	**Sensor-Data Gathering Software**
Reference to 3.1	Section 3.1.5.6 Big data, data fusion, Mine.io store and supply chain
Reference to 3.2	All UCs
Requirements	**Description:** Code for gathering mA from different sensors, creating a hex payload and send it to a lorawan cloud server for further decoding
	Operating System: The code was written in the data gathering component
	Input: mA
	Output: Hex payload

Table 54: Top-Down Requirements for SW Component "Data gathering from sensors" (INN)

SW Component	**Elasticsearch**
Reference to 3.1	Section 3.1.5.6 Big data, data fusion, Mine.io store and supply chain
Reference to 3.2	All UCs
Requirements	**Description:** Elasticsearch is a distributed search and analytics engine that is built on top of the Apache Lucene library. It is designed to handle large amounts of data and provide fast and flexible search functionality. Elasticsearch is commonly used for various purposes such as full-text search, log analytics, business analytics, and real-time data analysis.
	Operating System: Docker Linux Container / Linux

Table 55: Top-Down Requirements for SW Component "Elasticsearch" (INN)

12. Hellenic Mediterranean University (HMU)

HMU is currently investigating the communication protocols and open-source frameworks that are the most to meet the requirements of Task 5.4. The technical functionalities of the open-source software components that were investigated are shown in the tables below:

- EdgeX
- Arrowhead
- FIWARE
- Kura
- IoTivity

Utilised communication protocols:

- AMQP: Advanced Message Queuing Protocol
- MQTT: MQ Telemetry Transport
- RFC 7252: Constrained Application Protocol (CoAP)
- DDS: Data Distribution Service
- XMPP: Extensible Messaging and Presence
- HTTPS: Hypertext Transfer Protocol Secure

SW Component	**EdgeX**
Reference to 3.1	Section 3.1.5.5 Remote sensing, edge computing, edge continuum / dementalized communication system
Reference to 3.2	Section 3.2.4 Use Case 3: Geochemical Mapping of Soils and Mining Wastes in Lavrion Historic Mine Site – Post Mining Environmental Management Section 3.2.6 Use case 5: Advanced Mobility (Autonomous and Electrified Vehicles) and Operational Excellence)
Requirements	**Description:** The open-source EdgeX framework provides a stable basis for security and manageability while facilitating interoperability between diverse devices and applications at the IoT edge.
	Key Principles: Pluggable and Extensible Software Architecture, loosely-coupled microservices, platform independence
	Real-time: Yes
	Run-time: monitoring
	Open Source: YES
	Resource accessibility: High
	Additional: Under investigation – we will select the appropriate framework based on end users' feedback, Pluggable and Extensible Software Architecture, loosely-coupled microservices, platform independence

Table 56: Top-Down Requirements for SW Component “EdgeX” (HMU)

SW Component	**Arrowhead**

Reference to 3.1	Section 3.1.5.5 Remote sensing, edge computing, edge continuum / dementalized communication system	
Reference to 3.2	Section 3.2.4 Use Case 3: Geochemical Mapping of Soils and Mining Wastes in Lavrion Historic Mine Site – Post Mining Environmental Management Section 3.2.6 Use case 5: Advanced Mobility (Autonomous and Electrified Vehicles) and Operational Excellence)	
Requirements	**Description:** Eclipse Arrowhead is a framework capable to build automation and digitalisation solutions. It is based on microservice and micro system architecture. The implementation platform includes an engineering process and associated tools and core microsystems. Well proven core microsystem, libraries and template code for application systems, support for model-based engineering using SysML and UML are available open source.	
	Key Principles: SOA, Local Automation Clouds	
	Run-time: Dynamic Orchestration and authorization, monitoring, and dynamic automation	
	Open Source: *Yes*	
	Message Patterns: Req/Repl, Pub/sub	
	Transport Protocols: TCP, UDP, DTLS/TLS	
	Security Manager: Authentication, Authorization and Accounting Core System	
	Standardization: ISO standards for security, authentication...	

Table 57: Top-Down Requirements for SW Component "Arrowhead" (HMU)

SW Component	**FIWARE**
Reference to 3.1	Section 3.1.5.5 Remote sensing, edge computing, edge continuum / dementalized communication system
Reference to 3.2	Section 3.2.4 Use Case 3: Geochemical Mapping of Soils and Mining Wastes in Lavrion Historic Mine Site – Post Mining Environmental Management Section 3.2.6 Use case 5: Advanced Mobility (Autonomous and Electrified Vehicles) and Operational Excellence)
Requirements	**Description:** FIWARE based in Open Standards and in Open-Source software platform components. They can connect together with third-party components, used to build entire platforms in order to support Smart Solutions.
	Real-time: Yes
	Run-time: Dynamically monitor, track, and manage,
	Open Source: YES

	Resource accessibility: High
	3rd party and Legacy systems adaptability: Yes
	Security Manager: authenticate, connect and configure devices

Table 58: Top-Down Requirements for SW Component “FIWARE” (HMU)

SW Component	**Kura**
Reference to 3.1	Section 3.1.5.5 Remote sensing, edge computing, edge continuum / dementalized communication system
Reference to 3.2	Section 3.2.4 Use Case 3: Geochemical Mapping of Soils and Mining Wastes in Lavrion Historic Mine Site – Post Mining Environmental Management Section 3.2.6 Use case 5: Advanced Mobility (Autonomous and Electrified Vehicles) and Operational Excellence)
Requirements	**Description:** Eclipse Kura is an IoT Edge Framework based on Java/OSGi. Kura offers API access to the hardware interfaces of IoT Gateways. It features ready-to-use field protocols (including Modbus, OPC-UA, S7). In addition, Eclipse Kura supports a web-based visual data flow programming to acquire data from the field in order to process it at the edge send it to the IoT Cloud Platforms through MQTT connectivity.
	Open Source: YES
	Resource accessibility: High

Table 59: Top-Down Requirements for SW Component “Kura” (HMU)

SW Component	**IoTivity**
Reference to 3.1	Section 3.1.5.5 Remote sensing, edge computing, edge continuum / dementalized communication system
Reference to 3.2	Section 3.2.4 Use Case 3: Geochemical Mapping of Soils and Mining Wastes in Lavrion Historic Mine Site – Post Mining Environmental Management Section 3.2.6 Use case 5: Advanced Mobility (Autonomous and Electrified Vehicles) and Operational Excellence)
Requirements	**Description:** IoTivity support device-to-device connectivity along many wired and wireless Internet of Things (IoT) devices. Also, it can securely connect to each other and to the internet.
	Operating System:
	API: Resource Orientated REST
	Real-time: Yes
	Run-time: CRUDN, Messaging, Discovery, Monitoring, and Maintenance
	Distribution: Distributed

	Open Source: YES
	Resource accessibility: High
	Transport Protocols: TCP, UDP, DTLS/TLS, CoAP
	3rd party and Legacy systems adaptability: YES
	Security Manager: device identity, authentication, provisioning, establishing network credentials, authorization and access control

Table 60: Top-Down Requirements for SW Component "IoTivity" (HMU)

13. Institute for Systems and Computer Engineering, Technology and Science (INE)

INE is utilising the following software components, the technical functionalities of which are shown in the tables below:

- UX1-Neo exploration software
- EVA underwater localization software

Utilised hardware components:

- UX1-Neo: Autonomous underwater robot for surveying flooded underground spaces
- EVA: Autonomous underwater vehicle for open pit mine operation
- UX1-Neo cage 1: Robotised transportation system for deployment and recover of the UX1-Neo through vertical shafts

SW Component	**UX1-Neo exploration software**
Reference to 3.1	Section 3.1.1.3 Underwater vehicle for in-situ exploration in flooded mines
Reference to 3.2	Section 3.2.7 Use Case 6.1: In-Situ Underwater Exploration Myon Telescope Prototype Technology Validation for Water-Filled Mines Section 3.2.8 Use Case 6.2: In-Situ Underwater Exploration Robot Technology Validation for Water-Filled Mines
Requirements	**Description:** On board exploration software for the AUV
	Operating System: Linux
	Input: Mission specification
	Output: exploration mission execution

Table 61: Top-Down Requirements for SW Component "UX1-Neo exploration software" (INE)

SW Component	**EVA underwater localization software**
Reference to 3.1	Section 3.1.1.3 Underwater vehicle for in-situ exploration in flooded mines
Reference to 3.2	Section 3.2.7 Use Case 6.1: In-Situ Underwater Exploration Myon Telescope Prototype Technology Validation for Water-Filled Mines Section 3.2.8 Use Case 6.2: In-Situ Underwater Exploration Robot Technology Validation for Water-Filled Mines
Requirements	**Description:** On board localisation software

	Operating System: Linux
	Input: navigation sensors raw data
	Output: AUV local and global Localisation

Table 62: Top-Down Requirements for SW Component "EVA underwater localization software" (INE)

14. Łukasiewicz Research Network – Tele and Radio Research Institute (Ł-ITR)

Ł-ITR is utilising the following software components, the technical functionalities of which are shown in the tables below:

- PIT System
- Froth images repository
- PIT Estimation Module

Utilised hardware components:

- Workstation: Data processing and analysis, ML learning process
- Sensor Vision system: Data supplier for ML process
- Test data for binary images of flotation froth surface: Image pixels value

SW Component	PIT System
Reference to 3.1	Section 3.1.3.3 Artificial Intelligence technology for monitoring and control of metal ore processing
Reference to 3.2	Section 3.2.3 Use Case 2: Digital Flotation System – Artificial Intelligence technology for monitoring and control of metal ore processing
Requirements	**Input:** Vision system
	Output: Filtered/converted input data, Vision system control
	Database: Sensor (vision) data with timestamp stored in the internal HDD
	Timestamp: Timestamp will be added to registered images

Table 63: Top-Down Requirements for SW Component "PIT System" (Ł-ITR)

SW Component	Froth images repository
Reference to 3.1	Section 3.1.3.3 Artificial Intelligence technology for monitoring and control of metal ore processing
Reference to 3.2	Section 3.2.3 Use Case 2: Digital Flotation System – Artificial Intelligence technology for monitoring and control of metal ore processing
Requirements	**Description:** Type of data sharing will be decided with KGHM, Ł-EMAG and AGH during the realization of project
	Operating System: Windows
	Input: Vision system data
	Output: Vision system data
	Input and **output Formatting**: BIN

Table 64: Top-Down Requirements for SW Component "Froth images repository" (Ł-ITR)

SW Component	PIT Predictive Module
Reference to 3.1	Section 3.1.3.3 Artificial Intelligence technology for monitoring and control of metal ore processing
Reference to 3.2	Section 3.2.3 Use Case 2: Digital Flotation System – Artificial Intelligence technology for monitoring and control of metal ore processing
Requirements	**Description:** A module of Photonic-IT System (PIT) predicting the content of metal in the flotation froth by means of data preprocessing methods and trained ML models.
	API: REST
	Input data and format: froth image
	Output data and format: predictive metal content of the froth
	Programming language and IDEs: R/Python
	Exposed Services: set input data, execute prediction, retrieve results, visualise results

Table 65: Top-Down Requirements for SW Component "PIT Estimation Module" (Ł-ITR)

15. Łukasiewicz Research Network - Institute of Innovative Technologies EMAG (Ł-EMAG)

Ł-EMAG is utilising the following software component, the technical functionalities of which are shown in the table below:

- PIT Determining Module

SW Component	PIT Determining Module
Reference to 3.1	Section 3.1.3.3 Artificial Intelligence technology for monitoring and control of metal ore processing
Reference to 3.2	Section 3.2.3 Use Case 2: Digital Flotation System – Artificial Intelligence technology for monitoring and control of metal ore processing
Requirements	**API**: REST
	Input data and format: froth image
	Output data and format: determined metal content of the froth
	CI/CD tools: Git
	Exposed Services: set input data, execute prediction, retrieve results, visualise results
	Programming language and IDEs: Python

Table 66: Top-Down Requirements for SW Component "PIT Determining Module" (Ł-EMAG)

16. Frontier Innovations (FRON)

FRON is utilising the following software component, the technical functionalities of which are shown in the table below:

The **Workflow Automation Engine (WAE)** is a technology stack comprised by three components:

- Business Process Manager (FRON)
- Smart Contracts Manager (GFT)
- Hyperledger Besu Blockchain Network (TEC)

C SW Component	**MINE.IO Workflow Automation Engine**
Reference to 3.1	Section 3.1.5.2 Workflow automation and smart contracts for processes digitization in mines and raw materials value chains
Reference to 3.2	(none)
Requirements	**Description:** The Workflow Automation Engine manages operational and business workflows within Mines and throughout raw material value chains. In detail, it 1) models and visualises business operation workflows, 2) enables the decentralised collection of permits and certifications that cannot be falsified, 3) allows the digital signing of multi-party business contracts and 4) automates placement of work-orders to external vendors.
	Operating System: Windows, macOS, Linux (solution is containerised using Docker)
	Input: Business process definitions, user tasks, service tasks, and events that need to be managed and executed within workflows (via Web Interface/API).
	Input Formatting: BPMN 2.0 (Business Process Model and Notation) for process definitions. JSON or XML format for task data and configuration settings. Form data in JSON format for user task inputs. Output: JSON (via Web Interface/API)
	Input Interface: Interactive canvas for designing BPMN diagrams with palette and properties panel
	Output Description: Process execution results, task completion status, process metrics, and logs.
	Output Formatting: JSON format for API responses, task data, and process metrics. XML format for detailed process instances and deployment responses. HTML for web-based management and monitoring interfaces.
	Output Interface: Web-based Cockpit for monitoring and managing process executions.
	Additional: Workflow automation engine for handling the execution of operational and business workflows within Mines and across raw material value chains. Domain experts will be able to specify the workflows capturing business processes information while simplifying cumbersome processes including permitting procedures, through a graphical user interface. The engine will be equipped with analytics functionalities providing the ability to track decisions and identifying decision bottlenecks. The component is extended by a decentralised

	governance module, comprised of the Smart Contracts Manager (Table 36) and the Hyperledger Besu Blockchain Network (Table 44). Functionalities providing the ability to track decisions and identifying decision bottlenecks. Moreover, functionalities for materials' tracking across the value chain will be integrated within the proposed blockchain supporting traceability.

Table 67: Top-Down Requirements for SW Component "MINE.IO Workflow Automation Engine" (FRON)

17. Unexmin GeoRobotics (UGR)

UGR is utilising the following software components, the technical functionalities of which are shown in the tables below:

- GEOREKA
- CloudCompare

Utilised hardware components:

- UX1-Neo: Autonomous underwater robot for surveying flooded underground spaces
- BlueROV2: ROV for basic surveying of flooded spaces (down to 100m)
- DeepTrekker DTG3: ROV for basic surveying of flooded spaces (down to 200m)

SW Component	**GEOREKA**
Reference to 3.1	Section 3.1.1.3 Underwater vehicle for in-situ exploration in flooded mines
Reference to 3.2	Section 3.2.7 Use Case 6.1: In-Situ Underwater Exploration Myon Telescope Prototype Technology Validation for Water-Filled Mines Section 3.2.8 Use Case 6.2: In-Situ Underwater Exploration Robot Technology Validation for Water-Filled Mines
Requirements	**Description:** Geological modelling software
	Operating System: Windows
	Input: 3D point clouds, images
	Output: 3D model

Table 68: Top-Down Requirements for SW Component "GEOREKA" (UGR)

SW Component	**CloudCompare**
Reference to 3.1	Section 3.1.1.3 Underwater vehicle for in-situ exploration in flooded mines
Reference to 3.2	Section 3.2.7 Use Case 6.1: In-Situ Underwater Exploration Myon Telescope Prototype Technology Validation for Water-Filled Mines Section 3.2.8 Use Case 6.2: In-Situ Underwater Exploration Robot Technology Validation for Water-Filled Mines
Requirements	**Description:** Point cloud processing software
	Operating System: Windows, Linux, macOS

	Input: 3D point clouds
	Output: 3D model

Table 69: Top-Down Requirements for SW Component "CloudCompare" (UGR)

18. University of Salamanca (USAL)

USAL is utilising the following software component, the technical functionalities of which are shown in the table below:

- Google/ AWS

Utilised hardware components:

- Cloud Compare: Open-source tool for processing point clouds
- QGIS: spatial data generation tool
- Potree: open-source viewer for massive visualization of 3D point clouds
- GRAPHOS: open source photogrammetric and computer vision software for passing from 2D images to 3D point clouds
- BLK2Go: Mobile LiDAR for Mapping and Recording of forest areas in 3D
- HSC-Sim software
- ACT Platform
- Screening Eagle platform

SW Component	**Google/ AWS**
Reference to 3.1	Section 3.1.4.2 Pre-Classification of Stockpile Materials for Other Uses Section 3.1.4.3 UAV - Derived Electromagnetics and Magnetics for Mineral Exploration and Waste Identification
Reference to 3.2	Section 3.2.4 Use Case 3: Geochemical Mapping of Soils and Mining Wastes in Lavrion Historic Mine Site – Post Mining Environmental Management
Requirements	**Description:** Storage, run docker, and cost of running GEE
	Cloud API: Given by provider. Google or AWS
	Available storage space: 40TB
	Available Processing Cores/Threads: 12 cores
	Available RAM: 256GB

Table 70: Top-Down Requirements for SW Component "Google/ AWS" (USAL)

19. Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V. (FhG)

FhG is utilising the following software components, the technical functionalities of which are shown in the tables below:

- VolumePlayerPlus (own design)
- FireFly (own design)
- Volex10 (own design)
- IgorPro
- KlassifikatorGUI (own design)

Utilised hardware components:

- Prototype (own design): Prototype of dual energy XRT-sorting system
- Prototype (own design: Prototype of a CT-System with the relevant components to conduct a computed tomography scan
- Prototype (own design): X-ray transmission scattering experiment to calibrate EDX-device
- Prototype (own design): X-ray reflection experiment, uses energy dispersive setup to measure diffraction of crystalline structures

SW Component	**VolumePlayerPlus**
Reference to 3.1	Section 3.1.2.1 Enhanced drill core analysis in-situ using X-Ray technologies Section 3.1.3.2 Smart sorting with Dual-Energy XRT
Reference to 3.2	Section 3.2.1 Use case 1.1: Mine.io I4.0 Asset Digitalization
Requirements	**Description:** Software to visualize and process Xray data
	Operating System: Windows
	Input: 2D and 3D Xray-data
	Output: Visualization of Xray-data and segmented parts of the scanned object

Table 71: Top-Down Requirements for SW Component “VolumePlayerPlus” (FhG)

SW Component	**FireFly**
Reference to 3.1	Section 3.1.2.1 Enhanced drill core analysis in-situ using X-Ray technologies
Reference to 3.2	Section 3.2.1 Use case 1.1: Mine.io I4.0 Asset Digitalization
Requirements	**Description:** Software to process Xray projections and create Xray 3D-volumes
	Operating System: Windows
	Input: Radiographic projections
	Output: reconstructed 3D volume of scanned object

Table 72: Top-Down Requirements for SW Component “FireFly” (FhG)

SW Component	**Volex10**
Reference to 3.1	Section 3.1.2.1 Enhanced drill core analysis in-situ using X-Ray technologies Section 3.1.3.2 Smart sorting with Dual-Energy XRT
Reference to 3.2	Section 3.2.1 Use case 1.1: Mine.io I4.0 Asset Digitalization
Requirements	**Description:** Software to control X-ray systems
	Operating System: Windows
	Input: control parameters of X-ray system components
	Output: 2D X-ray projections

Table 73: Top-Down Requirements for SW Component “Volex10” (FhG)

SW Component	IgorPro
Reference to 3.1	Section 3.1.2.1 Enhanced drill core analysis in-situ using X-Ray technologies
Reference to 3.2	Section 3.2.1 Use case 1.1: Mine.io I4.0 Asset Digitalization
Requirements	**Description:** Software to analyze scattering data
	Operating System: Windows
	Input: 2D scattering data
	Output: Material composition of analyzed object

Table 74: Top-Down Requirements for SW Component "IgorPro" (FhG)

SW Component	KlassifikatorGUI
Reference to 3.1	Section 3.1.3.2 Smart sorting with Dual-Energy XRT
Reference to 3.2	(none)
Requirements	**Description:** Software to classify objects in x-ray images (including AI methods) with a graphical user interface
	Operating System: Windows
	Input: 2D X-ray data
	Output: classified objects and blow map for sorting

Table 75: Top-Down Requirements for SW Component "KlassifikatorGUI" (FhG)

5 MINE.IO ARCHITECTURE

Mine.io architecture has been based on the dual approach, presented in chapter 2 above:

On the one hand, the top-down approach (presented in chapter 2) has taken into consideration the different relevant frameworks, applicable to the smart industry domain, proposing aspects relevant to the mining industry which have been introduced into the Mine.io architecture, including the following concepts which have been added into the Mine.io architecture:

- cloud/edge architecture,
- functions relevant to the mining industry,
- core industrial functionalities and their structuring into layers (e.g. communication, information exchange, integration and asset management),
- different types of networking capabilities and solution required in mining industry,
- data-driven logical view of the architecture.

On the other hand, a bottom-up approach has been employed to analyse the Mine.io project's solutions/components in order to extract requirements for the Mine.io architecture. In this regard, components and Use Cases has been closely studied and provided input to the overall Mine.io architecture's design.

The analysis of requirements of Mine.io solutions and components, analysed in chapters 3 and 4 above, have provided input to the Mine.io architecture's design. The key takeaways of these requirements, relevant to the Mine.io architecture, are summarized below:

- **Basic Mine.io architecture structure** given that the Mine.io system requires both applications for the implementation of mining-relevant operations and supplementary solutions for supporting activities, then the basic breakdown of the architecture should be divided into Mining Operations and Support Technology Solutions.
- **Mining-Stage Operations:** all mining-relevant solutions and components developed by Mine.io are clustered under the four main mining stage-operations (i.e. Ore Exploration, Ore Extraction, Mineral Processing and Waste Management), consisting of sub-stage operations (Smelting, Smart Processing, Recycling) and Supply Chain Operations (Asset Management, Product Optimization).
- **Support Technology Solutions:** all non-mining operations to be developed in Mine.io are grouped as solutions performing Support Activities (i.e. Predictive Maintenance, Impact Assessment) and Digital Infrastructure (i.e. Data Fusion/Management, Digital Twins, Communication).

In detail, the design and specification of the Mine.io architecture has received valuable input from the analysis of requirements of Mine.io solutions, applications and software/hardware components, which is summarized in Table 76 below.

Architecture Specification	Requirements from GA	Requirements from Use Cases

Clustering of Mine.io end-solutions into Mining Operations	4 main mining stage-operation and multiple sub-stage operations	Each Use Case consists of solutions categorized in specific mining phases
Groups of sub-stage operations as Support Activities to complement the solutions of Mining Operations	Supplementary technology solutions which support the execution of the mining-related operations	-
Supply Chain Operations provide monitoring/optimization services to the Mining Operations and enhance the Support Activities	Mining-related solutions/components which make part of the 4-stage mining operations	-
A Digital Infrastructure offers core services to all Mine.io solutions, e.g. connectivity/data management/DT	Solutions providing services to Mine.io components, providing horizontal services related to communication channels, data management etc.	-

Table 76: Input from Requirements into the Mine.io Architecture Specification

5.1 HIGH-LEVEL REFERENCE ARCHITECTURE WITH FUNCTIONAL DOMAINS

The second version of the Mine.io High-Level Reference Architecture (HLRA), illustrated in Figure 10 below, has been influenced by the dual approach HLRA which has been decided among the Mine.io partners in D2.2.

Both of the two High-Level Reference Architectures represent different layers of digital transformation in the framework of the Mine.io project, with a clear linkage between the physical, on-site mining operations and the advanced digital platforms. The second version of the HLRA introduces the **cloud-based evolution** of the first schema, focusing on how digital transformation elevates mining operations, as following:

1. **Real-World Mine Industry Operations (Real world Layer)**: The real-world mining activities such as exploration, ore extraction, mineral processing, and waste management are the same as those in the first diagram but shown here in a linear progression. This layer represents the actual physical processes that take place in mining operations.

2. **Digital Platform (Lower Pyramid Layer)**: Above the real-world operations, the digital platform, including technologies such as big data storage, cybersecurity, and remote sensing come into play. These technologies provide the digital backbone for gathering and processing data from the real-world layer (physical mining operations). This layer aims at optimizing mine industry operations by enabling real-time data acquisition and analysis using sophisticated technologies.

3. **Mine.io Cloud (Top Pyramid Layer)**: The top of the pyramid depicts the Mine.io cloud, which is responsible for providing sophisticated digital services such as predictive maintenance, social sustainability, workflow automation, and PLM/SDM modules. These supportive solutions allow the mining operations to be digitally integrated with advanced management and optimization tools. The Mine.io cloud serves as a central platform for overseeing and clustering the mining operations (exploration, ore extraction, mineral processing and waste management), the Support Activities (cloud services), the Supply Chain Operations (combination of tasks and cloud services), and the Digital Infrastructure (connectivity, data management, and digital transformation services).

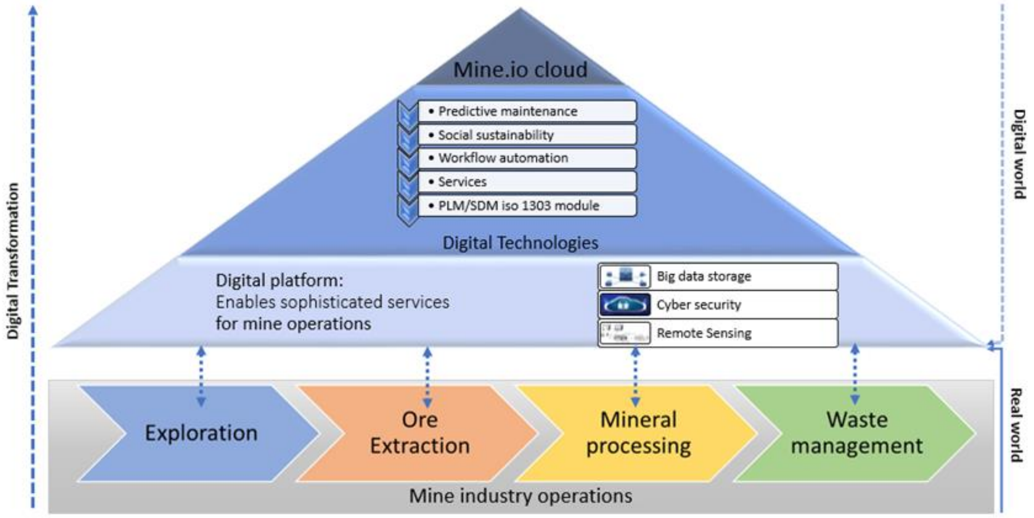

Figure 10: High-Level Reference Model for Digital Mining Operations and Functions

5.2 Logical View

The first version of the Mine.io architectural logical view, presented in D2.2, offers a three-tiered framework of all the components to be utilized in Mine.io, encompassing mining stage operations, supportive value-added solutions, supply chain operations, Mine.io digital infrastructure, asset management, and cybersecurity applications. It introduced the physical layer, where data are gathered from the mining environment, the data management and processing layer, where dataflow is handled and processed for advanced data analytics, and the mining digital twins' layer, where predictive maintenance and optimization processes have been taken place. Finally, the cyber security crosses the entire architecture, ensuring data protection from the field devices to the digital twins, and the cloud services.

At this deliverable, we are going to introduce the second version of the logical view of the Mine.io architecture as depicted below in Figure 11. As the project is progressing, a variety of multiple needs and requirements, either from technology providers' perspective, or from the pilots' one, are coming to surface, influencing our conceptual point of view. The evolution of this view focuses on the convergence of diverse technologies aimed at improving several stages of the mining operations' lifecycle—from exploration and ore extraction to mineral processing and waste management, integrating the digital technologies, the cloud services, the data processing systems and the business mining workflows to the Mine.io platform.

At this version of the Mine.io logical view, we are going to introduce an abstract overview of how the multiple components of the Mine.io platform are integrated as to offer sophisticated mining operations management. That abstract layered approach consists of the following layers, where each one performs a distinct role, ranging from data acquisition at the field to cloud-based analytics and decision-making. An insightful description of the core layers and their interaction follows.

1. Field Layer

The Field Layer plays a crucial role in the Mine.io logical view, consisting of various physical components such as sensors, machines, and other smart devices, creating an advanced communication network. All those devices, including equipment such as drills, conveyor belts, semiautonomous driving trucks and sorting systems, are utilized to gather data from the field, either in synchronous or asynchronous way, and 'feed' into the cloud for further processing. The field layer is part of mine enterprise.

2. Work Cell Layer

The two key features of the work cell layer are modularity and interoperability. At this layer each discrete operational component represents a specific service or a set of diverse tasks, facilitating the communication, synchronization and data exchange. The work cell layer includes both software and hardware components that are able to handle functionalities such as equipment control, automation processes, data preprocessing, and decentralized decision-making. In a typical mining environment, work cells can include automated drill systems, sorting machines, or mining oriented equipment. The data gathered from the field layer is processed at this layer, and critical decision-making, such as real-time operational modifications, can be made here before sending data to the pilot level. This layer also belongs to the mine enterprise.

3. Pilot Layer

The Pilot Layer is of fundamental importance, operating as an intermediary between the mine enterprise and the overarching Mine.io platform, and transforming the mining processes into smart, data-driven operations. The primary functionality of this layer focuses on data aggregation from various work cells and, simultaneously, leveraging AI and ML optimization processes, enabling remote control and real-time, or near to real-time data visualization and management. Its key components include gateways and IoT devices that facilitate communication between the individual local systems and the Mine.io platform, ensuring efficient data transmission through edge computing and remote sensing technologies. Finally, this layer often provides initial data preprocessing 'feeding' with the proper data upper layer.

4. Mine.io Platform

The Mine.io Platform acts as the core at this schema, housing advanced data processing. This platform integrates various technological solutions, performing data analytics, predictive maintenance, and AI/ML based decision-making. Some major functionalities include the predictive maintenance module, and other services such as the workload scheduling, the workflow automation (including smart contracts), and the blockchain-based traceability systems. Furthermore, the big data platform plays a vital role as the integration point between the southbound layer and the northbound services, 'consuming' and 'producing' data respectively, expediting to be utilized either as a storage solution, or as an asset cataloguing module for the Pilots that already possessed their own ones. Finally, we should not forget the Cybersecurity and Identity and Access Management (IAM), being an essential component of this layer, providing security mechanisms, ensuring platform integrity and resilience.

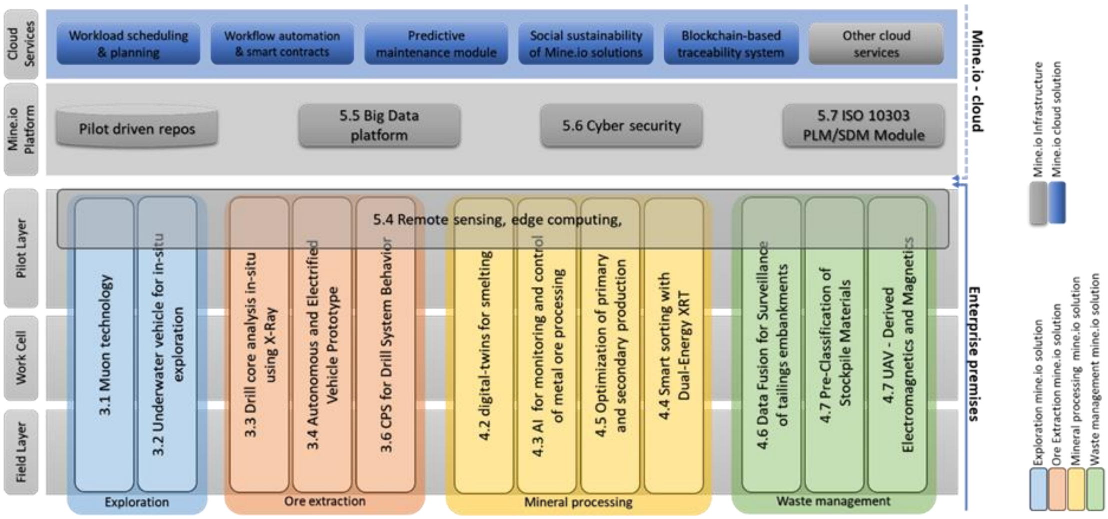

Figure 11: Logical View of Mine.io Architecture

5.3 IMPLEMENTATION VIEW

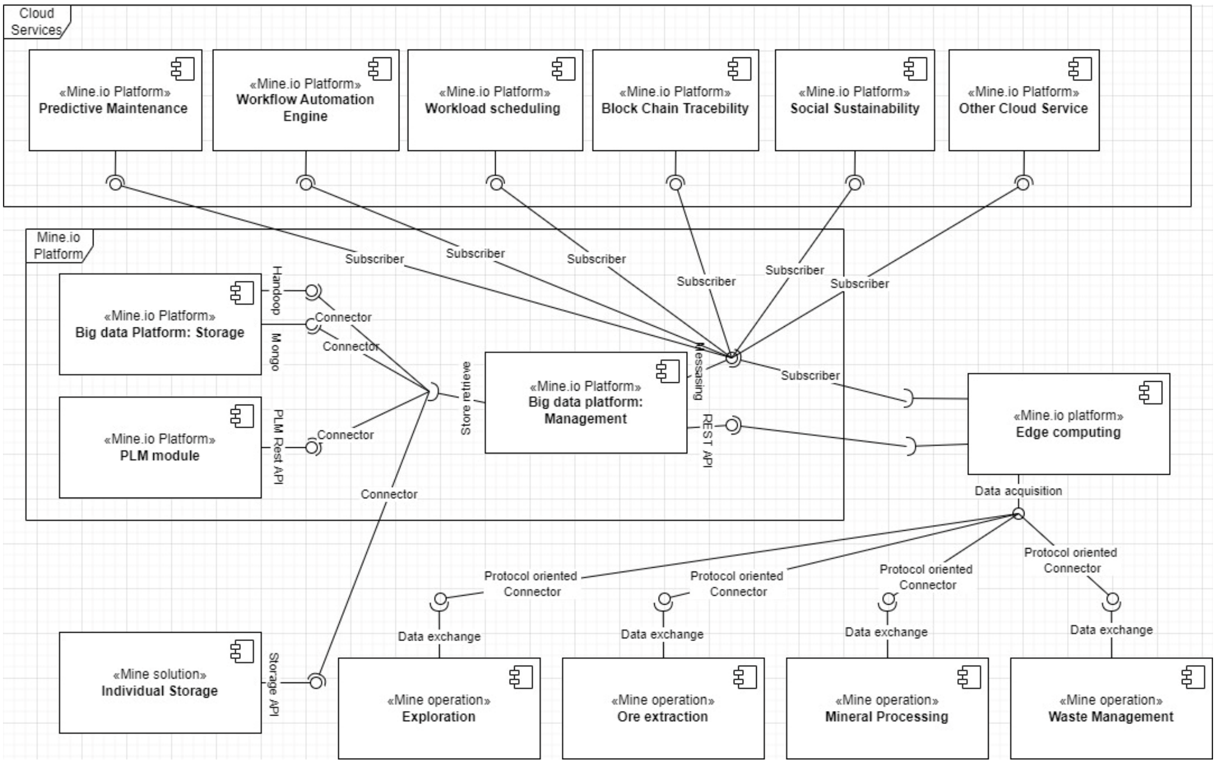

Figure 12: Implementation view (component diagram)

The Implementation View section outline the Mine.io from development point of view. focusing on its software components and their interdependencies. In Figure 12 Mine.io platform components are presented. The top section presents cloud services that consume data from the Big Data platform. Messaging communication patterns have been established (Kafka messaging system is selected) for data retrieval. Cloud services instantiate a set of subscribers/consumers in order to consume data. For the data storage as depicted in the middle section of the figure, a set of connectors are defined, depending on the storage solution. A variety of storage solutions is supported based on pilots' requirements. To the bottom side, Edge Computing is responsible for data acquisition from mine operations and for sending data to the Big Data platform. The communication between the mine operations and the edge computing is protocol driven. A variety protocol implementation can be supported such as OPC-UA, MQTT, UDP etc. Finally, the data exchange between edge and the big data platform can be served in two ways: a) asynchronous if we deal with real time streaming or b) synchronous if we deal with batch data. Here we should mention that instead of presenting mine operations as single components, their internal structure is more complex and also, consists of a combination of physical machineries, sensors, control and process units. All the above-mentioned information is presented in an abstraction way, highlighting only the functionality that is related with data exchange. More details for the internal structure of all components will be presented to the deliverables of next work packages, WP3 and WP4 for mine operations and WP5 for Mine.io platform components.

5.3.1 Security Components and design

Cybersecurity is a critical aspect of any modern system, and Mine.io, focus on creating cyber-secure mines, is no exception. This section outlines the cybersecurity requirements and strategies for Mine.io, aiming to protect the system's confidentiality, integrity, and availability.

Organizations must comply with regulatory frameworks like GDPR and the EU Directive on Critical Entities Resilience. These regulations often require balancing security measures with privacy concerns, presenting a challenge for effective cybersecurity. This is considered in the architecture.

Cybersecurity involves a comprehensive set of technologies, practices, and measures aimed at preventing and mitigating the effects of cyberattacks. The primary goals of cybersecurity are to protect three core properties of secure systems: confidentiality, integrity, and availability.

Confidentiality: This involves ensuring that sensitive data is not disclosed to unauthorized individuals. Techniques to uphold confidentiality include authentication, authorization, access controls, and encryption. A breach in confidentiality occurs when sensitive information is exposed or leaked to unauthorized parties.

Integrity: Maintaining integrity means ensuring that data and system files remain unaltered without proper authorization. Compromises in integrity might involve tampering with logs or modifying access controls, potentially leading to unauthorized changes in data.

Availability: This property ensures that systems and data are accessible to authorized users when needed. Threats to availability include attacks that cause system downtime or prevent legitimate users from accessing services.

Mine.io is in the deployment phase of Keycloak, an open-source Identity and Access Management (IAM) solution. The decision to use Keycloak is made due to its widespread support of existing systems, including Jotne's EDMtruePLM™, which is used as a backend solution in the Mine.io architecture. Keycloak which provides a centralized platform for managing user identities, authentication, and authorization across multiple applications, enhancing security by protecting against unauthorized access and data breaches. Additionally, Keycloak improves the user experience with Single Sign-On (SSO) and streamlined authentication processes, while also increasing efficiency through automated user provisioning and deprovisioning. Keycloak's modular architecture enables customization and integration with other systems, making it a versatile and adaptable solution for various use cases such as Mine.io's. Future work will focus on identifying and addressing security gaps in Mine.io architectures to enhance overall cybersecurity effectiveness. Furthermore, work with privacy (and by extension confidentiality) preserving technologies (PPT), especially focussed on Access Control will be performed to facilitate the integration of results into a zero-trust architecture.

5.4 PROCESS VIEW OF MINE.IO ARCHITECTURE

In order to design the process view of the architecture, sequence diagrams have been drafted, which show the process interactions arranged in time-order for an end-to-end execution of complete functions. Sequence diagrams depict the participating actors, the processes and objects involved, as well as the sequence of messages exchanged between processes/objects required to carry out the software's functionalities.

5.4.1 Process view of Big Data Platform

This section describes the functionality provided by the Big Data Platform. Figure 13 and Figure 14, depicts the sequence diagrams for message-oriented communication. REST API is used only for the data asset manipulation (discover an existing or creating a new). When the producer or consumer receives the connector details, instantiates a Kafka client and publishes or consumes data.

Figure 15 and Figure 16 depicts the sequence diagrams for data exchange through REST API. REST API initially manipulates the appropriate data asset operation (discover or register a new one) and after that selects the appropriate connector in order to perform the data exchange operation (store or retrieve). The selected connector performs the exchange with the data storage solution.

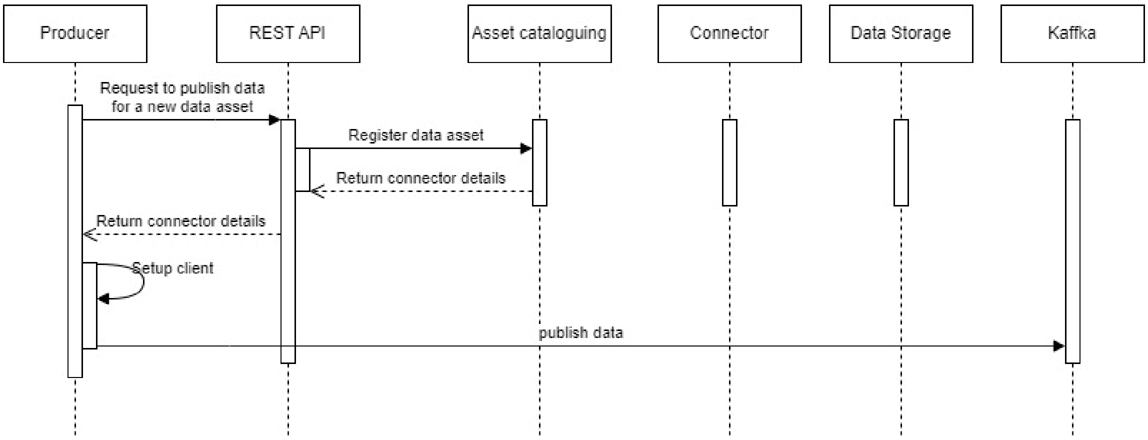

Figure 13: A data producer registers a data asset and publishes real time streaming to the messaging system.

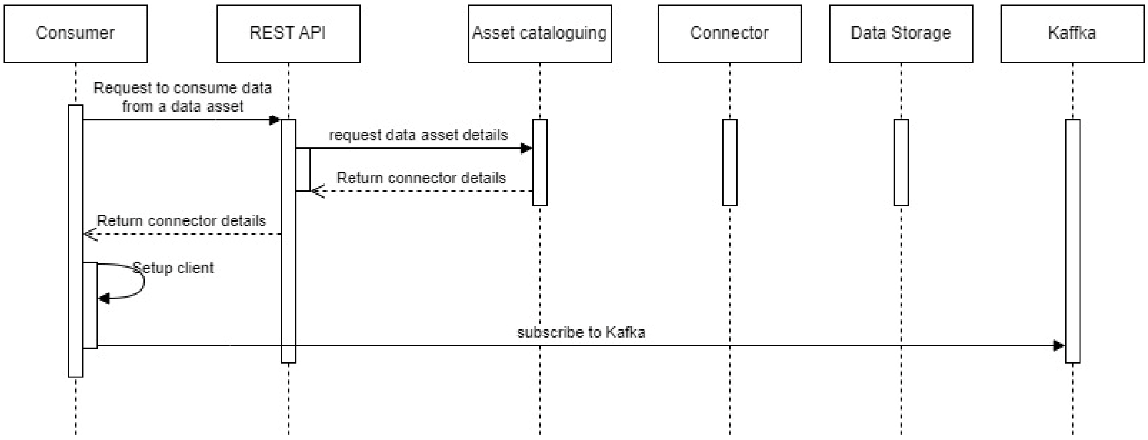

Figure 14: A data consumer discovers a data asset and subscribes to messaging system to consume the real time streaming.

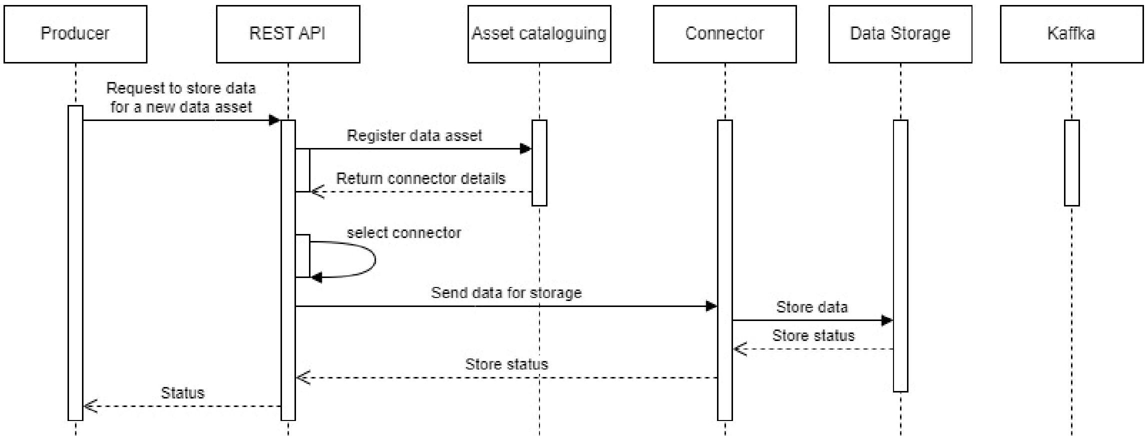

Figure 15: A producer registers a new data asset and stores data through REST API

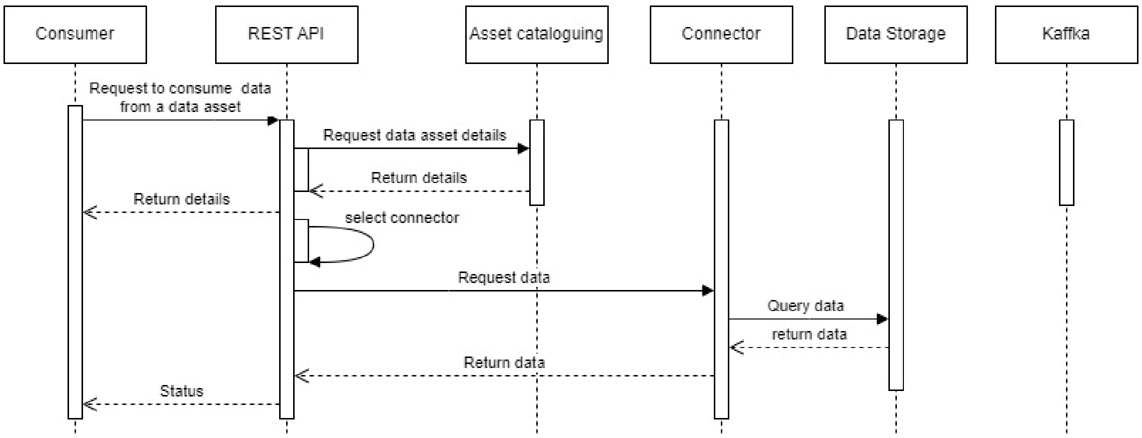

Figure 16: A consumer requests to retrieve data from a data asset through REST API

Finally, Figure 17 and Figure 18 is an extension of functionality that is provided by the previous functions. Similarly with the previous sequence diagrams, the REST API performs the operations that are related with data asset manipulation (discover an existing or creating a new). However, the actual data exchange is not performed by the REST API but directly between connector and producer or consumer.

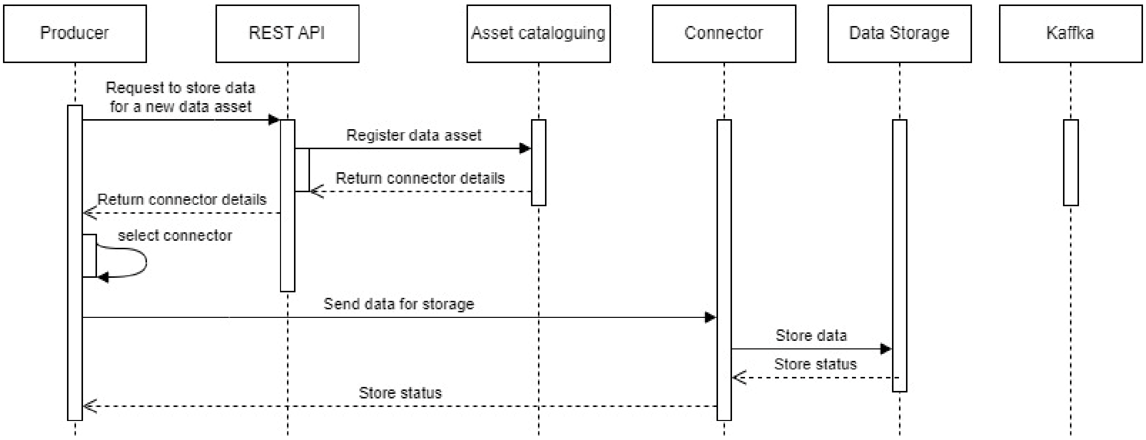

Figure 17: A producer registers a new data asset and stores data directly to the storage solution.

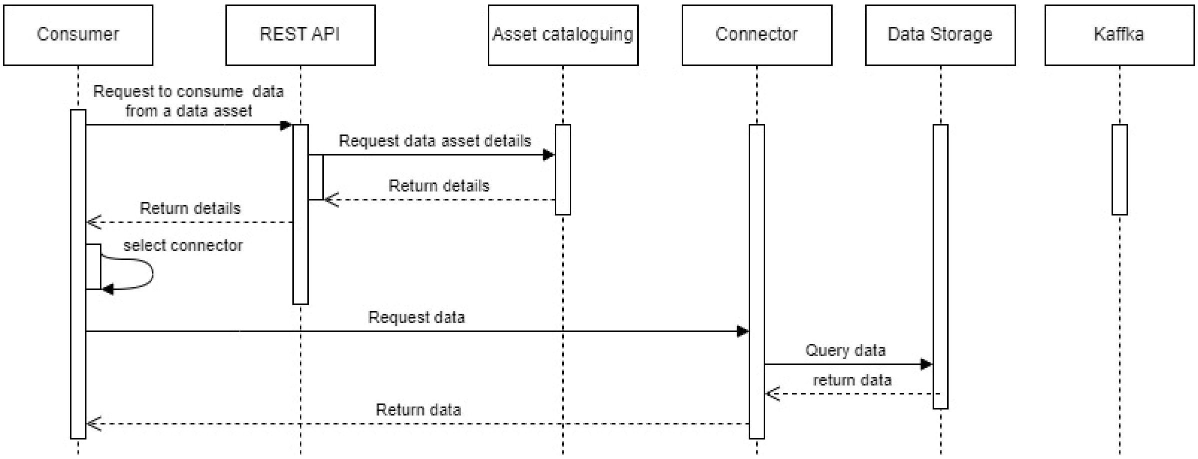

Figure 18: A consumer requests to retrieve data from a data asset directly from the storage solution.

5.4.2 Process Views of other components

In this section, indicative sequence diagrams of different Mine.io components and systems are presented, which are used in Pilots and Use Cases. This is not an exhaustive list, since the sequence diagrams of all Mine.io components and Use Case scenarios will be presented in the next version of the deliverable, while the sequence diagrams of below-listed components will be enhanced and their more consolidated views will be presented in subsequent versions of the deliverable, where better aligned to the overall Mine.io logical architecture will be illustrated.

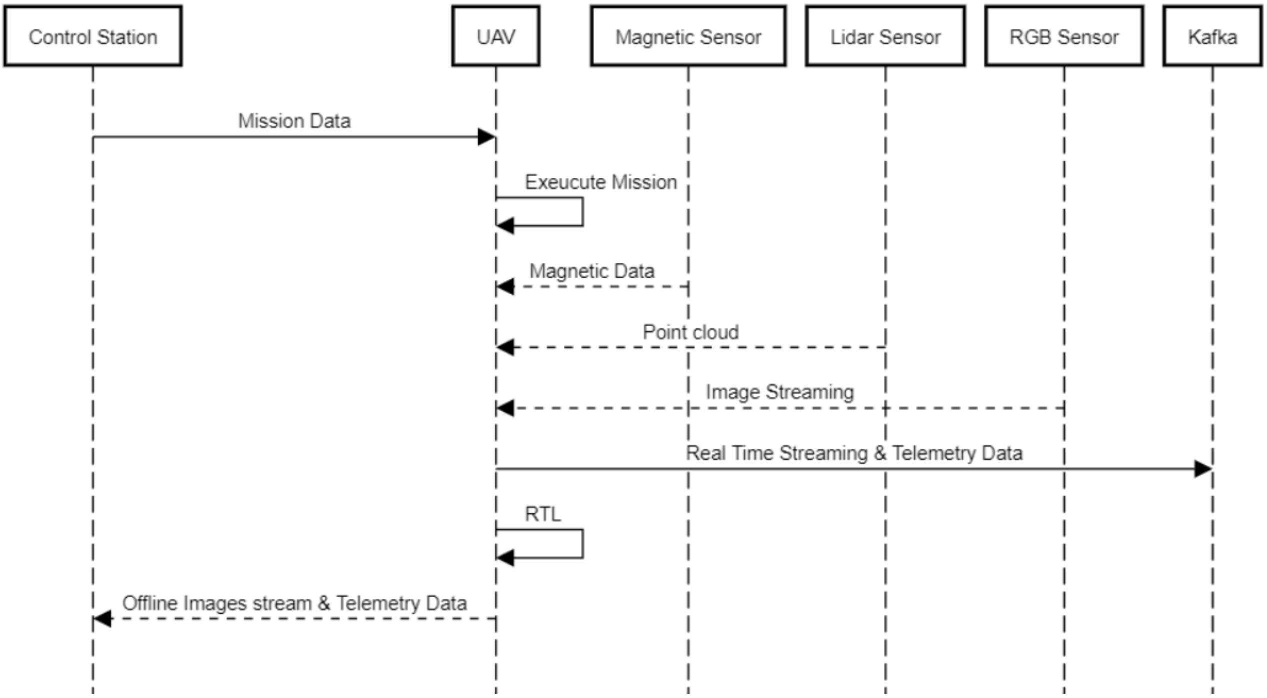

Figure 19: Sequence Diagram for Cerberus UAV (ACC)

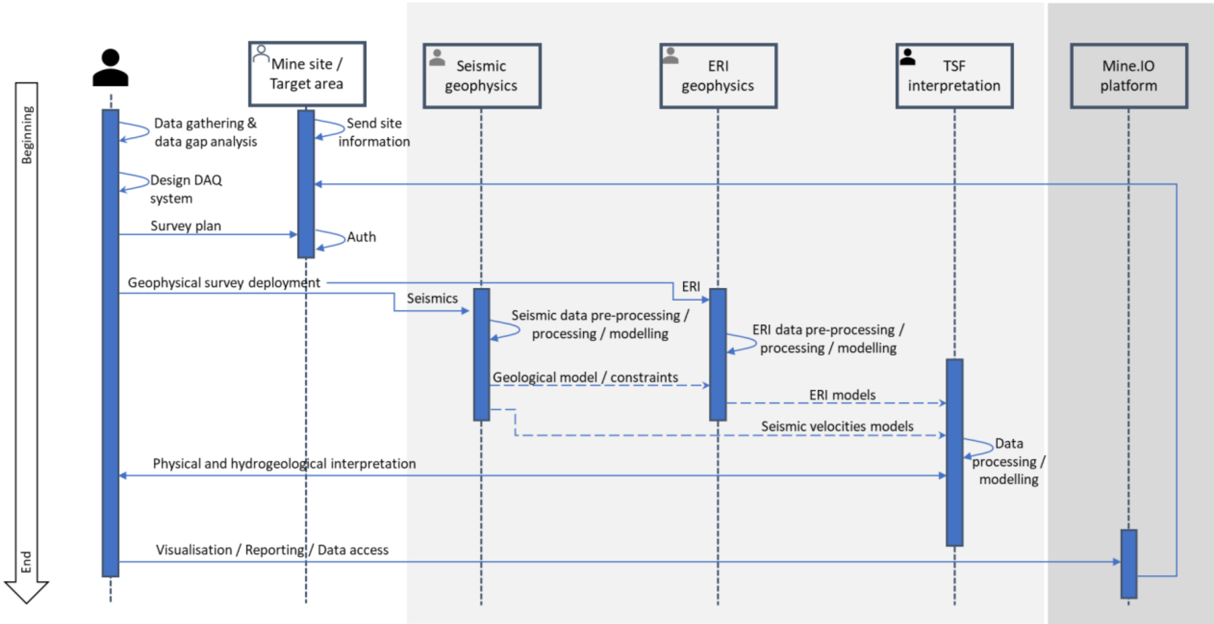

Figure 20: Sequence Diagram for Subsurface-Conditions Interpretation Tool (UOULU)

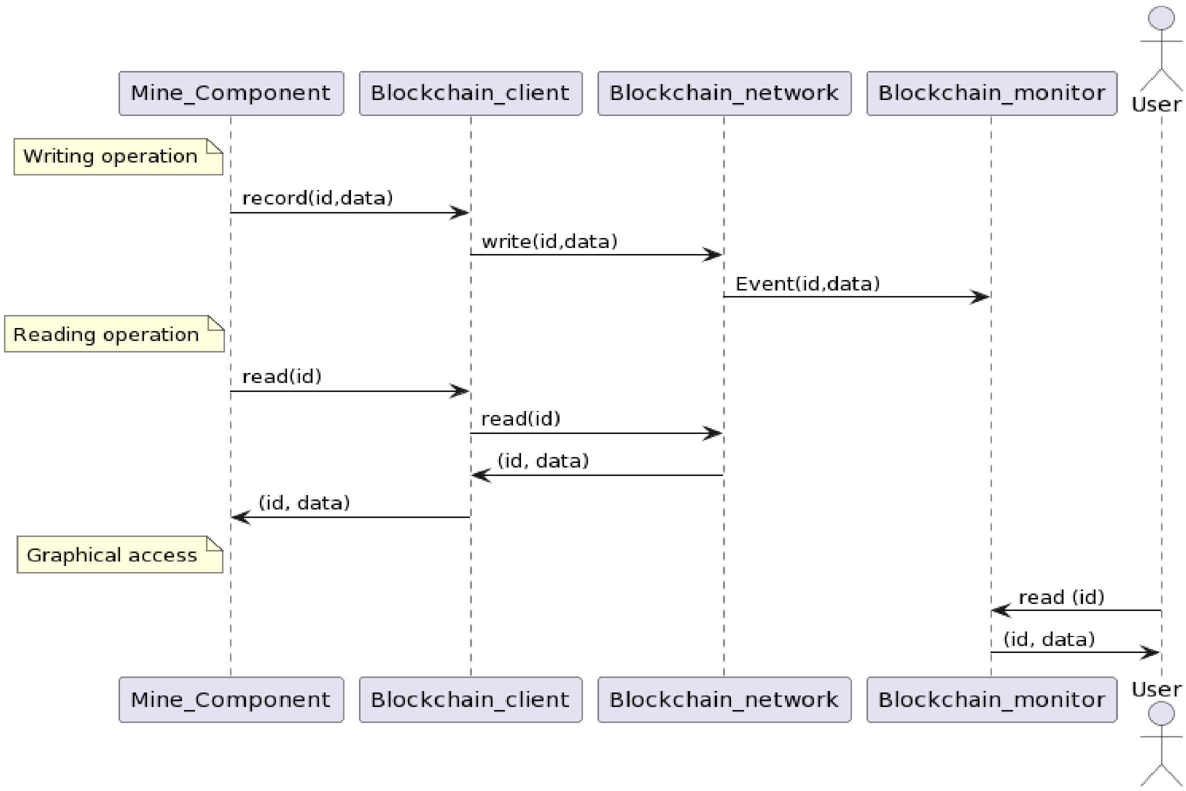

Figure 21: Sequence Diagram for Blockchain-based Traceability Tool (TEC)

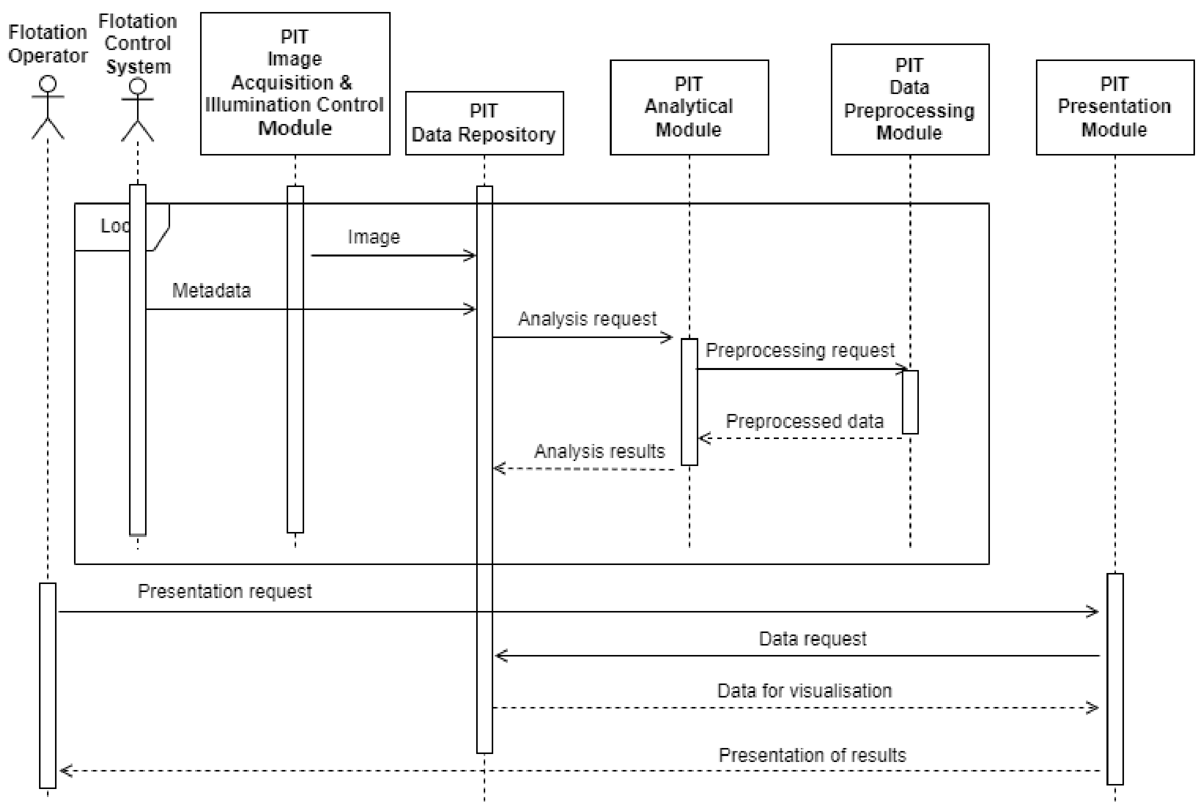

Figure 22: Sequence Diagram for Photonic-IT Component (L-ITR)

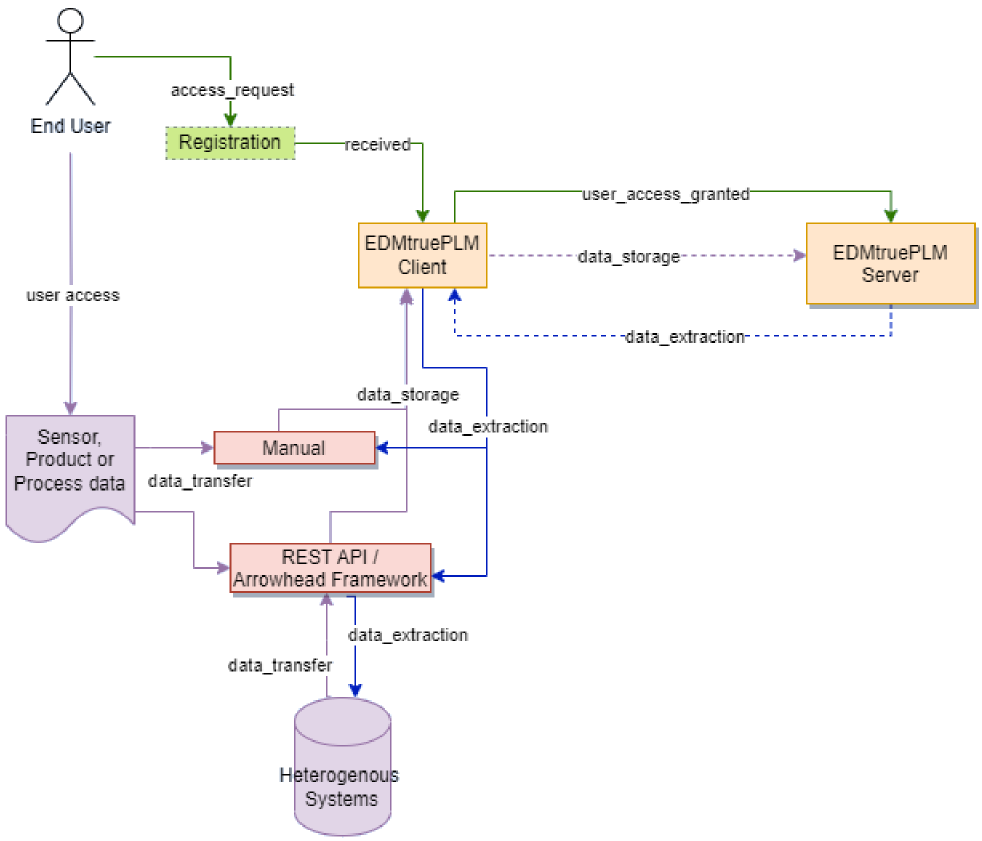

Figure 23: Sequence Diagram for EDMtruePLM/ISO 10303 Repository Component (JOT)

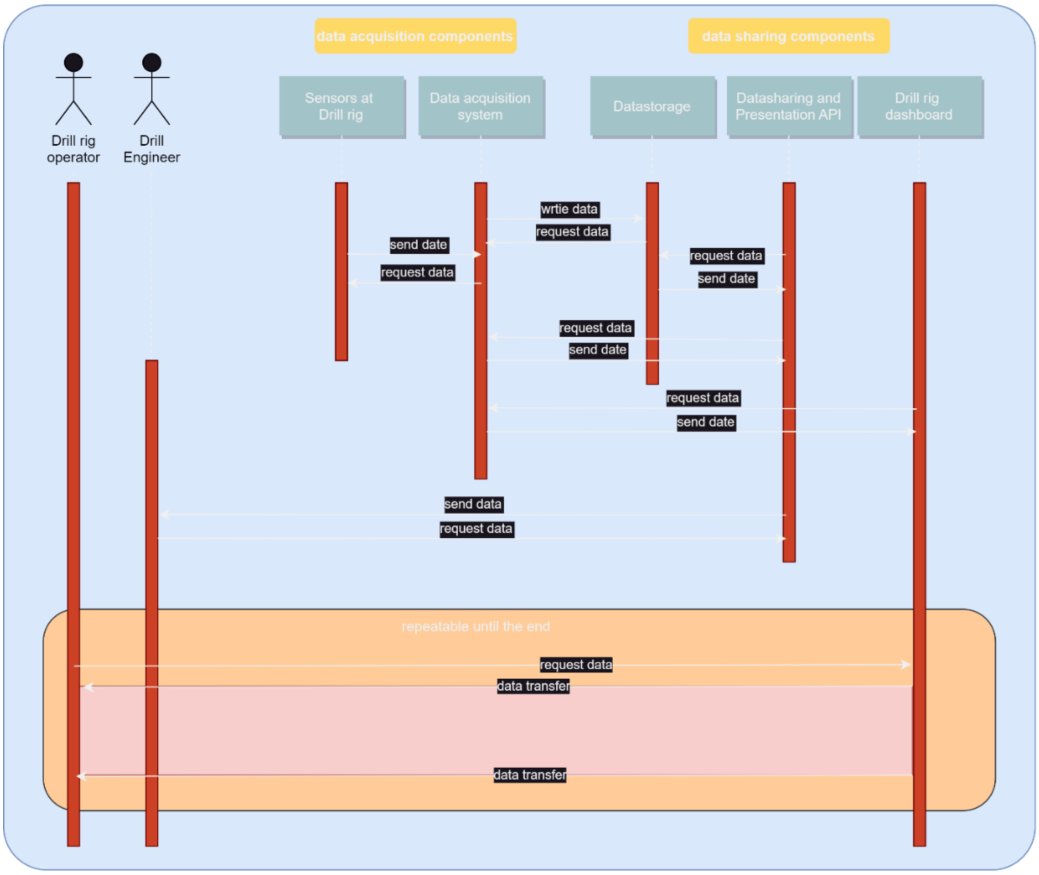

Figure 24: Sequence Diagram for Drill Rig Operation Component (TUB)

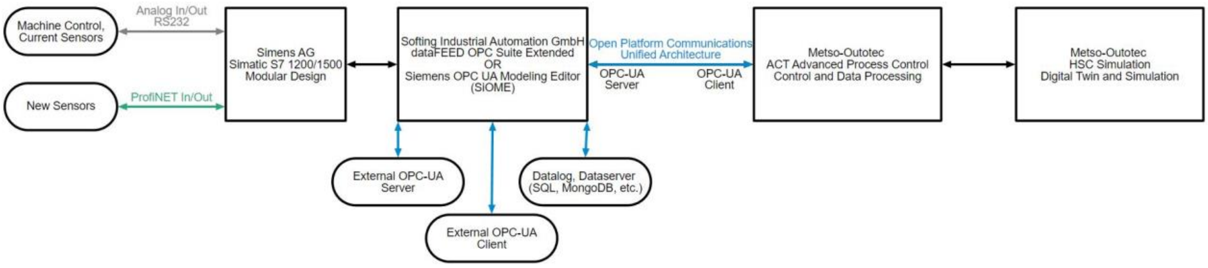

Figure 25: Sequence Diagram for Digital Smelter Component (TUB)

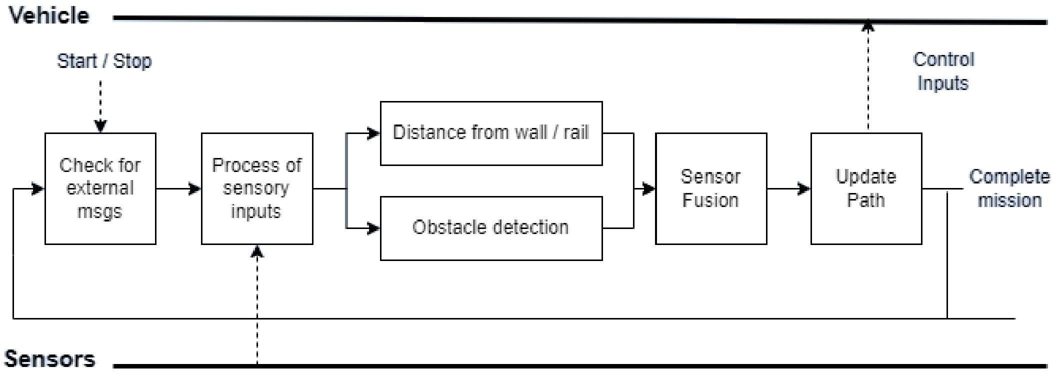

Figure 26: Sequence Diagram for ANIM Component (ACC)

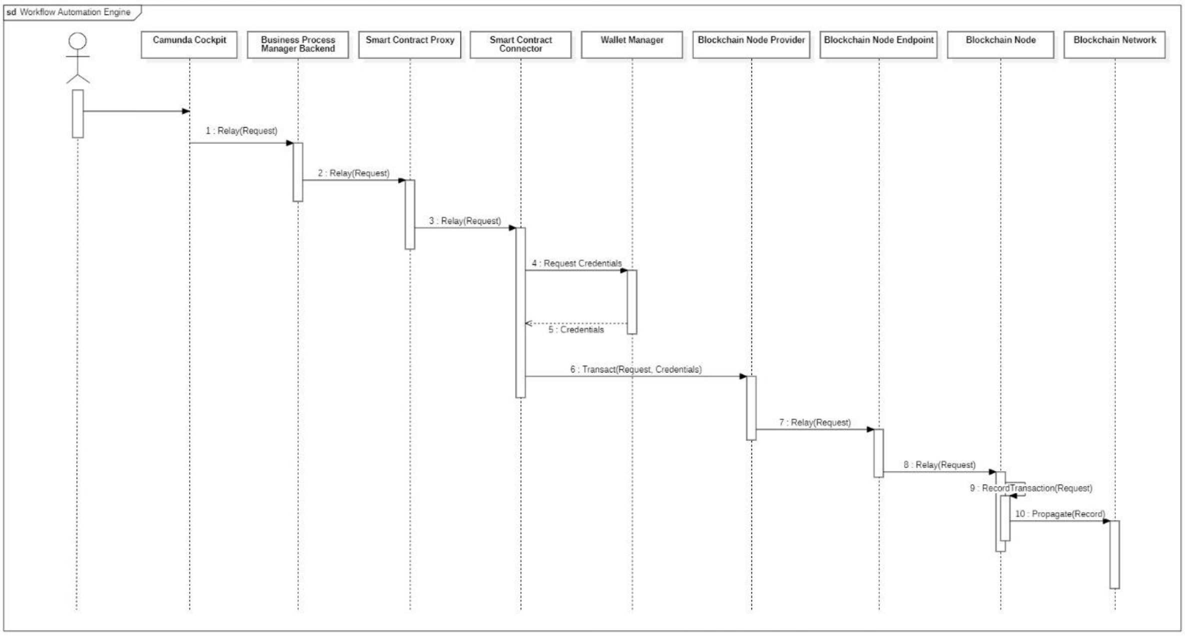

Figure 27 Sequence diagram for Workflow Automation Engine(FRON)

5.5 DEPLOYMENT VIEW

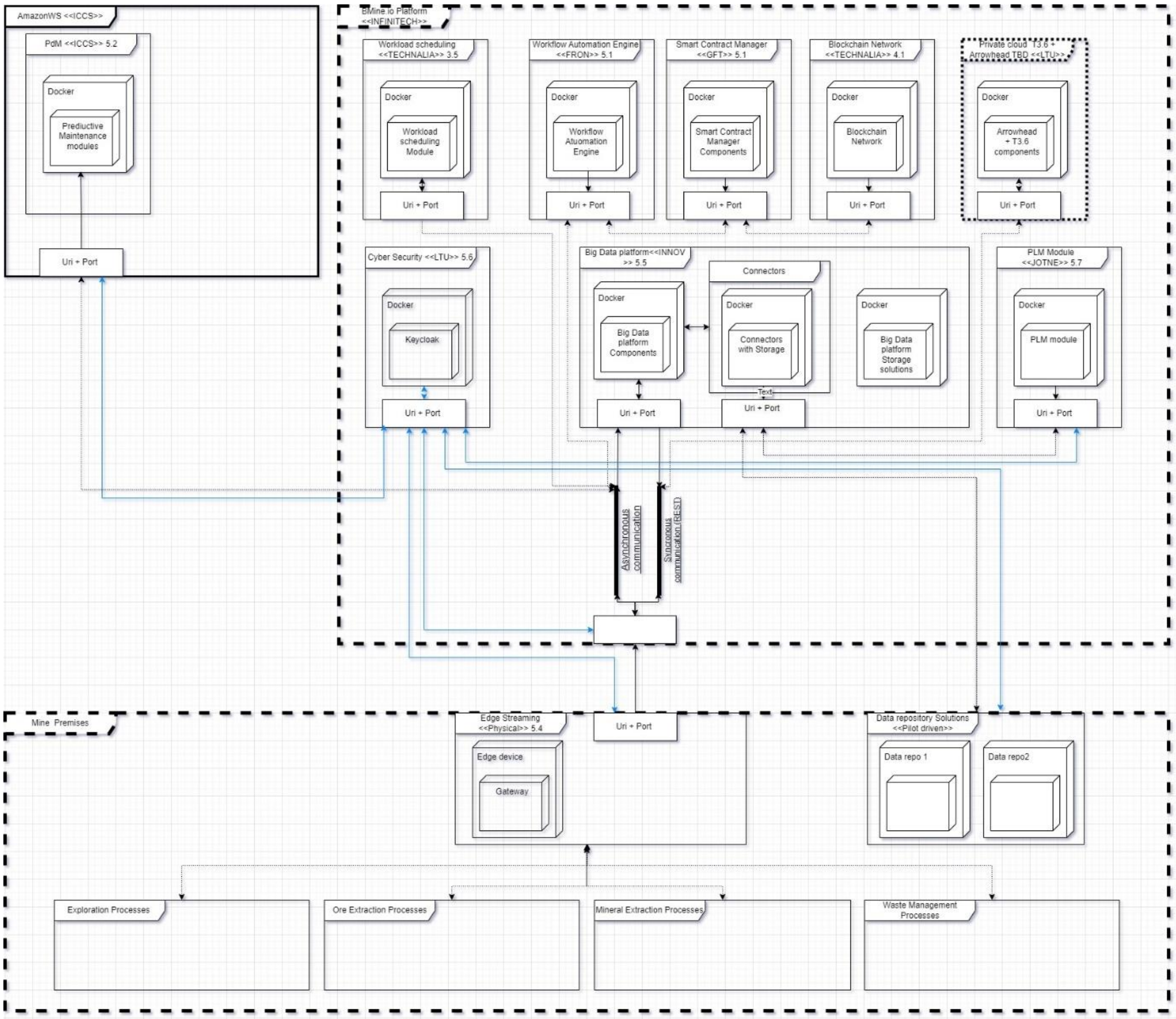

Figure 28 Deployment diagram

Figure 28 shows the initial version of the deployment diagram. The Big Data platform, as part of the Mine.io platform, serves as the central orchestrator within the system, managing and coordinating the flow of data, services, and communications between the various components of the architecture. By using a combination of asynchronous (Kafka), and synchronous (REST APIs), communication channels, it ensures efficient and reliable operation of the entire system, maintaining data integrity, service interoperability, and compliance with industry standards. Acting as a hub for service integration, the Big Data platform connects to multiple modules, including the PLM module, predictive maintenance, workflow automation Engine, workload scheduling etc, that comprise the cloud services part of Mine.io platform. From the southbound of the Mine.io platform there is the connectivity with external systems (pilots) via the Edge Streaming module. The Edge Streaming module collects data from the pilots and transmits it to the Mine.io Big Data platform, enabling real-time processing and analysis. The INFINITECH platform was defined as deployment infrastructure, which has been described in chapter 7.

Since the components of the Mine.io project will be deployed in a shared digital infrastructure for the Use Case scenario implementation, their implementation and deployment technical details serve as driving parameters for the customization of the Mine.io architecture.

In this regard, Table 77 below presents key implementation and deployment information per component, which will be taken into consideration for the configuration of the Mine.io CI/CD infrastructure and will provide considerable input to the architecture.

Component	Partner	Language	API (REST, none or Socket)	OS	Docker	Gateway/Edge or Edge/Cloud or Cloud
Muon telescope	WRCP	Python	Socket	Linux	N/A	Gateway/Edge Edge/Cloud
Underwater vehicle for in situ exploration	INE	C/C++ Python	No	Linux	N/A	Edge
AUV-derived 3D imaging	INE	C/C++ Python	No	Linux	N/A	Edge
X-ray scattering and computed tomography	FhG	N/A	No	Windows	No	N/A
Electrification of underground trucks	IPT	C Python	No	Windows	N/A	Edge/Cloud
Autonomous driving module	ACC	C++ Python	No	Linux	N/A	Edge
Workload scheduler and planner	TEC	Java JavaScript HTML Python	REST API	Linux	Yes	Cloud
CPS for system drill behaviour	LTU	Matlab	No	Windows	No	N/A
Blockchain-based traceability system	TEC	Solidity JavaScript	REST API	Windows Linux	Yes	Edge/Cloud
Sensor based-digital-twins-for-smelting	TUB	OPC-UA C/C++/.NET	REST API	Windows	Yes	Gateway/Edge
Photonic-IT (PIT) System	L-ITR	Delphi C#	REST API	Windows	Yes	Gateway/Edge
Image processing and analysis	L-ITR	Delphi C# R Python	REST API	Windows	Yes	Gateway/Edge
Dual energy XRT for smart sorting	FHG	N/A	No	Windows	No	N/A

Primary and secondary production Optimizer	LTU	MatLab FMU	FMI	Portable	No	Edge/Cloud
Multi-source data fusion and Interpretation for Surveillance of tailings embankments	UOULU	Python	REST API	Windows	AWS or Azure	Gateway/Edge
3D Modelling & Pre-classificator of stockpiles materials and electromagnetics	USAL	Phyton C++	REST API	Windows Linux	N/A	Cloud (WebGIS)
Workflow automation engine	FRON	Java	No	Windows, macOS, Linux	Yes	Cloud
Smart contract manager	GFT					
Predictive maintenance module	ICCS	Python	GraphQL API	Windows Linux	Yes	Cloud
Social Life Cycle Assessment	POL	COM interface .NET	No	Windows	No	Gatewat/Edge
Remote sensing, edge computing, edge continuum / dementalized communication system	HMU	Python Java	Resource Oriented REST	Windows Linux	No	Edge/Cloud
Big data Manager	INN	Java C#/React	REST API	Linux	Yes	Edge/Cloud
Techniques for security and privacy of data exchanged	LTU	Java	REST Arrowhead	Agnostic	Yes	Edge/Cloud
PLM/SDM Module	JOT	EXPRESS VUE.js	REST Arrowhead	Windows	No	Cloud

Table 77: Implementation and Deployment Details of Mine.io Components

6 SCENARIOS-SPECIFIC ARCHITECTURES

Following the proposed Mine.io architecture presented in chapter 5, subsequent architectures can be designed based on the solutions, components, and hardware devices to be utilized in each of the project's Pilots and Use Case scenarios.

The sections below show the current state of each Use Case's architecture from the 4+1 logical view, highlighting its detailed features while adhering to the overall Mine.io architectural approach. These scenario-specific architectures are all based on the Mine.io architecture and have been detailed to include all the unique components to be put into practical use and tested during the Use Cases.

It is important to note that the Use Case architectures are in an early stage, and changes and improvements are anticipated during the iterative implementation of each Use Case within the Mine.io architecture. This iterative process will focus on optimizing data flow by continuously refining how data is collected, processed, and transmitted to enhance efficiency and performance. Additionally, it will involve enhancing communication to improve the methods and protocols used for communication between components, ensuring seamless integration and operation. Feedback from initial deployments will be incorporated to fine-tune data handling and communication strategies. Finally, it will continuously enhance security measures to protect data and ensure compliance with relevant regulations.

6.1 PILOT 1:1 FREIBERG MINE, GERMANY (TUBAF)

Mine.io project's Pilot 1.1 will be carried out in Freiberg Mine, Germany the components/solutions and architecture is illustrated in Figure 29 below. Pilot 1.1 involves installing sensors with a data logger on an existing core drilling rig, followed by evaluating the data using AI to focus on drill behaviour and predictive maintenance. Additionally, a digital twin of the mine ventilation system will be developed, which includes reviewing and installing multiple sensors. Workflow automation is also considered in this pilot.

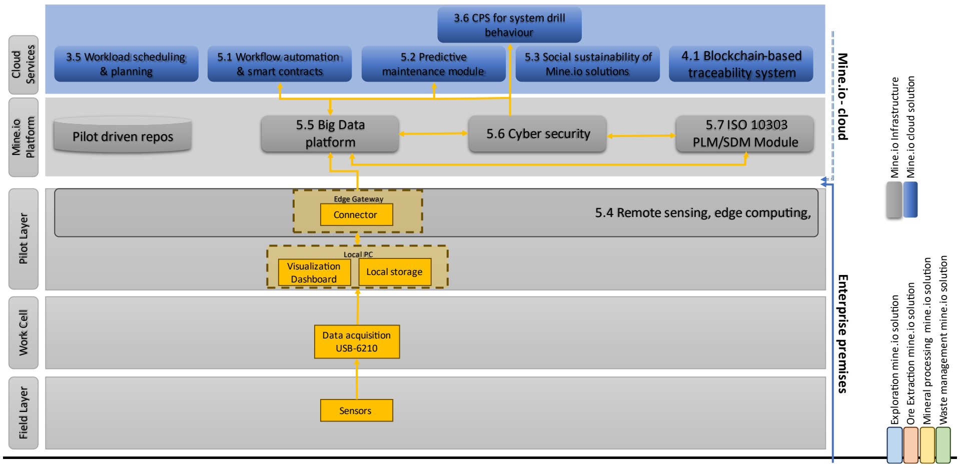

Figure 29: Scenario-Specific Architecture for Use Case 1.1

The various data management tasks are performed by different partners. The big data platform from the project partner INN serves as the central data distribution platform. Data management, including the storage of all current and historic data, takes place on the PLM module of the project partner Jotne. Access control and identity management for the data handled through the big data platform will be performed by LTU within T5.6. The project partner FRON is developing a workflow automation tool in task T5.1. The resulting data calculated by this tool will be visualized by a web-based dashboard application. The AI algorithms of the partners ICCS and LTU for analysis are connected via the big data platform and can access current and past data at any time. Interfaces between all these applications are Apache Kafka, REST-APIs and the Arrowhead framework.

6.2 Pilot 1:2 Freiberg Smelter, Germany (TUBAF)

Mine.io project's Pilot 1:2 will be carried out in Freiberg, Germany and consists of Use Case 1:2 scenario, the components/solutions and architecture of which are illustrated in Figure 30. Pilot 1.2 is situated at the Institute of Nonferrous Metallurgy in Freiberg. The Top Submerged Lance (TSL) smelter will showcase the application of various sensor techniques, integrated with simplified embedded models derived from Computational Fluid Dynamics (CFD) calculations and online process models (digital twins).

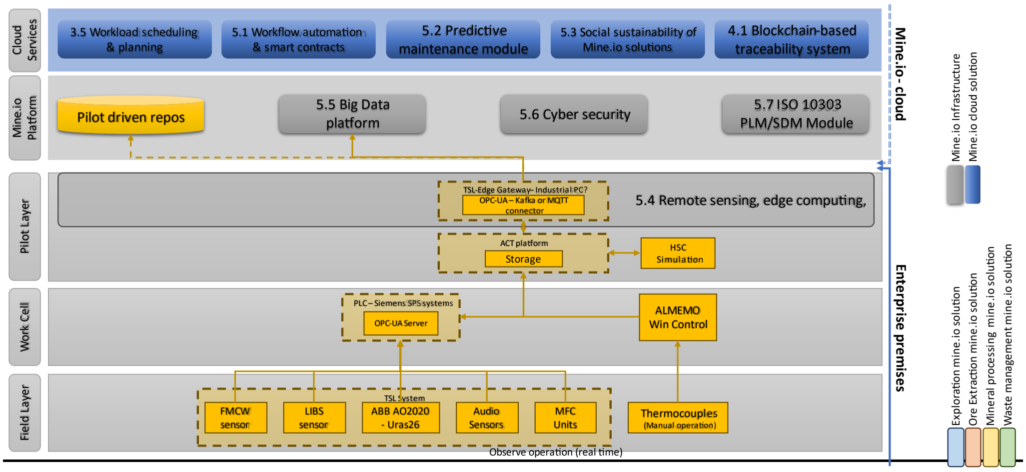

Figure 30: Scenario-Specific Architecture for Use Case 1.2

All Data are gathered real time, except temperature measurements inside the furnace, which will be done manually by using a lance and therefore have a slight delay until input into system. Data Collection Components in the field layer deliver the raw data and are thereby the first and base layer on which all the data processing relies on. Especially important for process handling are input flow rates of the primary energy source and gas as well as the analysis of the off-gas via FTIR in combination with the temperature, which goes directly into calculations of the reactive atmospheric conditions in the process. For data transmission from sensors to Siemens SPS system, real time data protocols in analogue and digital are used. For older sensors like thermocouples and simple sensors for the lance height, valve positions and off-gas throughput, analogue communications via voltage differences or RS 232 are used. Further sensors are communicating via industrial standard communication protocol ProfiNet for highly reliable IP-Network data transmission. Inside SPS, all relevant data streams get converted into open-source real time protocol, OPC-UA data and distributed to the Edge Gateway. The local

Edge Gateway collects all important data, handles filtering, processing and calculations of different process variables like lambda value of the kiln atmosphere. Software wise we use a combination of an OPC-UA server to handle all data flows and the ACT Advanced Process Control for processing and giving all relevant parameters to the digital twin. From our Edge Gateway we can synchronize and send data via MQTT Brokers, OPC-UA protocol or to SQL server. Also receiving data streams is possible via databank input, MQTT publisher or OPC-UA stream. The data flow inside the pilot plant and to the outside, gives an overview about the used communication routes, protocols and devices. The combination of the SPS connected via OPC-UA with the PC running ACT and HSC represents our Gateway Edge system connected to the Mine.io platform. Communication between different clients and server will be secured with up-to-date encryption and 2-factor authentication systems. The OPC-UA protocol uses state of the art standardized secure encryptions and authentication features for maximum security. All incoming data streams from sensors as well as from Mine.io platform will get prepared, processed, filtered, and data structure wise united. For modelling in and within our digital twin, HSC-Sim will simulate flow sheet simulation and show all relevant live parameters of the plant as well as calculated ones in a user-friendly format for technical staff. All parameters including the calculated ones will get sent forward to our Cloud as a structured and usable data stream combinable with our material and process parameters as well as the timecode belonging to the data sets.

6.3 PILOT 2 POLKOWICE, POLAND (KGHM)

Mine.io project's Pilot 2 will be carried out in Polkowice (Poland) and consists of Use Case 2 scenario, the components/solutions and architecture of which are illustrated in Figure 31. The pilot focuses on PIT System development and validation in the flotation process. A Photonic-IT (PIT) System will be installed and tested on one of the flotation cells in this plant.

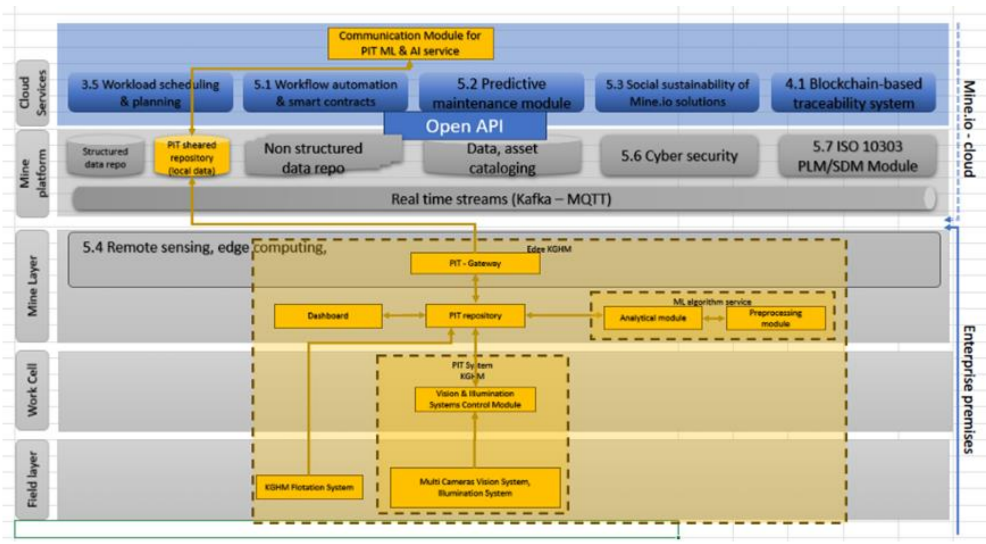

Figure 31: Scenario-Specific Architecture for Use Case 2

The current architecture is illustrated in Figure 31 and relationship is as follows. The PIT - Gateway in the edge layer acts as the primary interface for data flow between the lower layers (Work cell and Field layer) and the higher layers (Mine.io platform and cloud services). The

Dashboard is used for visualization and control, pulling data from the PIT repository. The PIT repository serves as the central data storage, feeding data to the Analytical module and the Preprocessing module. The Preprocessing module processes raw data before it's analyzed by the Analytical module. The PIT system controls the vision and illumination systems, which are part of the work cell layer, and collects data from the Multi Cameras Vision System and Illumination System. The KGHM Flotation System at the field layer sends operational data to the PIT system for further processing and control. Due to the confidentiality of process data and KGHM policy, communication with Mine.io platform will be restricted.

6.4 PILOT 3 LAVRION, GREECE (AMDC)

Mine.io project's Pilot 3 will be carried out in Lavrion (Greece) and consists of Use Case 3.1 and Use Case 3.2 scenario, the components/solutions and architecture of which are illustrated in Figure 32. The Use Case 3.1 objective is to conduct a thorough analysis of the dry tomb (mining/metallurgical waste repository) to optimize site characterization and extract valuable information on soil composition and distribution. This data will be used to develop a feasible metal extraction method and/or a suitable management or remediation strategy for pilot application (waste management). Use Case 3.2 refers to an Undeground Hazardous Waste Repository, and represents an additional demonstration during the evolvement of the project, and it is not described in the DoA or the GA. The Partners considered that it is an ideal site for implementing the Muon Technology (T3.1) targeting in the exploration stage. Use Case 3.2 objectives include area characterization and obtaining detailed data on the ground composition and distribution. Additionally, the density distribution of the volume above the muon detector is being measured. This data may also provide insights into the repository structure, such as identifying fractures or faults, which could help address issues like water leaks and generally poor rock quality.

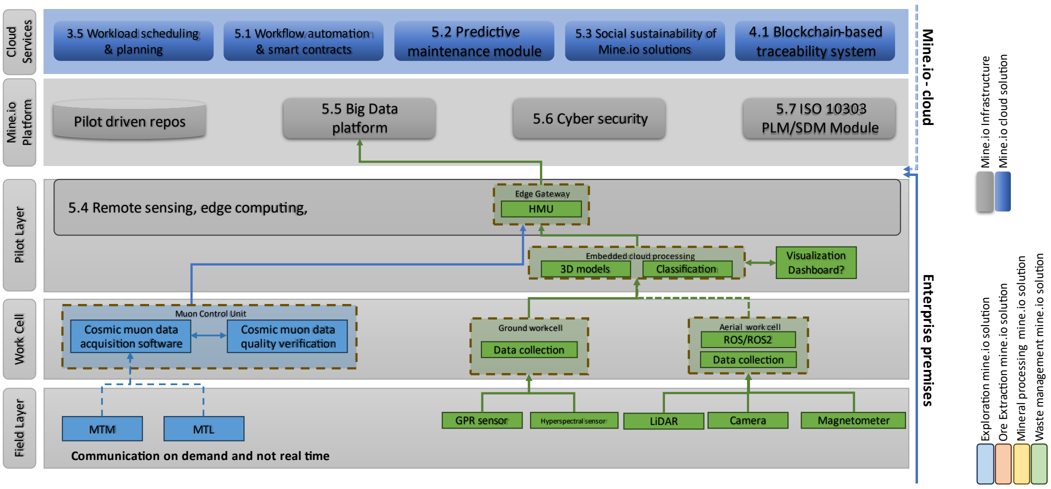

Figure 32: Scenario-Specific Architecture for Use Case 3

The current architecture is illustrated in Figure 32 and relationship is as follows. Data from field layer sensors (Muon, GPR, hyperspectral, LIDAR, cameras, magnetometers) are collected by the Muon Control unit, Ground work-cell, and Aerial work-cell, respectively. These data are processed at the edge before being sent for further analysis. Edge processing involves data aggregation, preliminary filtering, and the execution of localized decision-making algorithms. Embedded cloud processing handles 3D modelling and data classification, utilizing inputs from various data collection modules. To support field personnel, a Visualization Dashboard tailored to non-technical users is used. This interface displays processed data and actionable insights in an accessible format, enabling on-the-ground teams to make informed decisions quickly. The Edge Gateway acts as a local server within the Lavrion Pilot's architecture, bridging the gap between the Mine.io platform and the array of field data collectors. Its primary role is to manage and streamline field-level operations, ensuring smooth data flow from the edge of the network to core processing facilities. The Mine.io platform incorporates a dedicated pilot-driven repository for storing operational logs, processed data, and other relevant information. The Big Data Platform performs analytical processing on raw data to extract insights.

6.5 Pilot 4 Pyhäsalmi, Finland (Oulu)

Mine.io project's Pilot 4 will be carried out in Pyhäsalmi (Finland) and consists of Use Case 4 scenario, the components/solutions and architecture of which are illustrated in Figure 33. The pilot will develop and implement a geophysical approach, combining electrical resistivity imaging (ERI) and both active and passive seismic imaging, to assess the subsurface condition of tailings embankments.

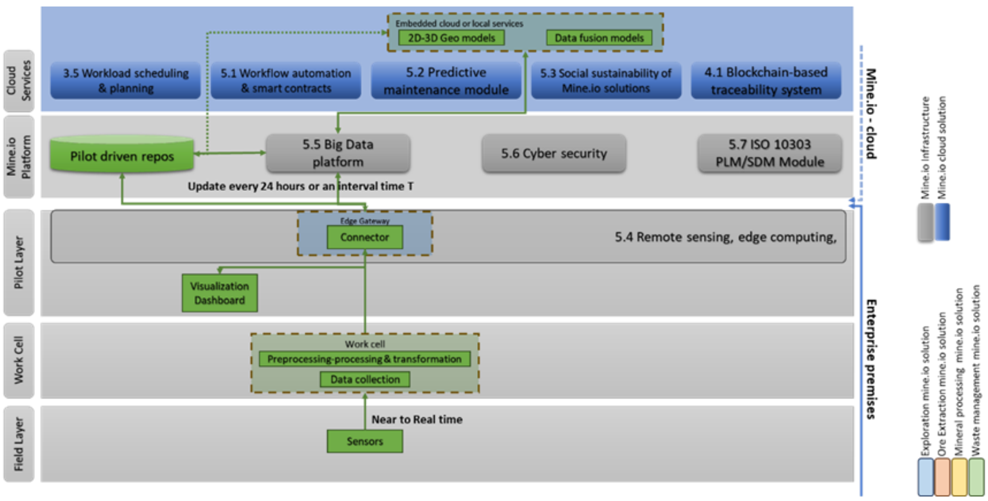

Figure 33: Scenario-Specific Architecture for Use Case 4

The current architecture is illustrated in Figure 33 and relationship is as follows. Real-time data is continuously gathered by sensors in the Field Layer, which capture information from the tailings. The raw data is transmitted to the Work Cell Layer, where it undergoes initial processing and transformation, refining it into a format suitable for further analysis. From here, the processed data is passed to the Edge Gateway in the Pilot Layer, which acts as a bridge, securely sending the data to the Big Data Platform in the Mine.io Platform Layer. This platform serves as a central repository, storing and managing the data for accessibility. The Pilot Driven Repo is synced with the Big Data Platform, ensuring that they always contain the latest information. The sensor data is utilized by advanced modelling tools like 2D-3D Geo Models and Data Fusion Models in the Cloud Services Layer, which perform complex analyses to generate insights and enhance decision-making. Finally, the analysed data and insights are presented through the Visualization Dashboard in the Pilot Layer, where they are made accessible to users, allowing them to monitor current state of the tailings.

6.6 PILOT 5 NIEDERSCHLAG, GERMANY (EFS)

Mine.io project's Pilot 5 will be carried out in Niederschlag (Germany) and consists of Use Case 5 scenario, the components/solutions and architecture of which are illustrated in Figure 34 below. The pilot aims to test underground mobility solution that integrates electric-driven battery vehicles and autonomous driving functions in underground trucks. This will involve integrating electrification paradigms, developing necessary charging parameters, and implementing an automatic docking system for inductive charging.

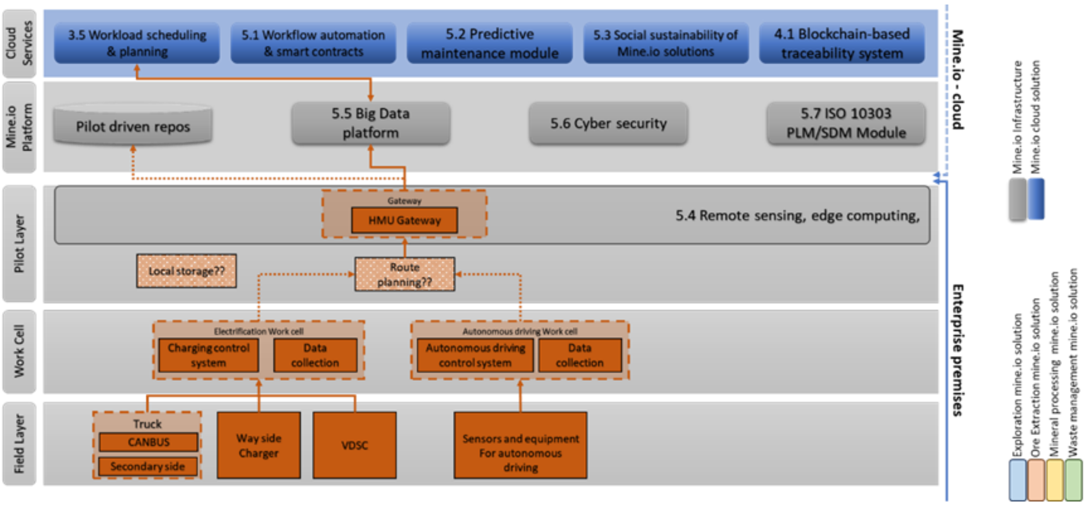

Figure 34: Scenario-Specific Architecture for Use Case 5

The current architecture is illustrated in Figure 34 and relationship is as follows. The architecture is intended to optimize the charging process by managing it while simultaneously controlling the truck's autonomous driving. The collected data from the sensors is transmitted to the Big Data Platform and the Pilot driven repo in the Mine.io Platform Layer, where it is processed and analysed using cloud services such as Workload scheduling and planning to create more efficient work schedules. Furthermore, the cloud services Workflow automation could automate routine tasks, freeing up human workers for more complex tasks. Once the data has been analysed and processed, instructions and commands are sent to the Field Layer. Such as optimized routes to the truck, control its autonomous driving, and manage the charging process.

6.7 PILOT 6 MALAPOSTA AND URGEIRICA , PORTUGAL, (INE)

Mine.io project's Pilot 6 will be carried out in Malaposta and Urgeirica (Portugal) and consists of 2 similar but distinctive Use Case scenarios (Use Case 6.1 and Use Case 6.2), the components/solutions and architecture of which are illustrated in Figure 35 and Figure 36. Use Case 6.1 will conduct testing and validation of a muon imaging and monitoring instrument designed for underwater mining applications. Use Case 6.2 will test an underwater vehicle specifically engineered for in-situ exploration in flooded mines.

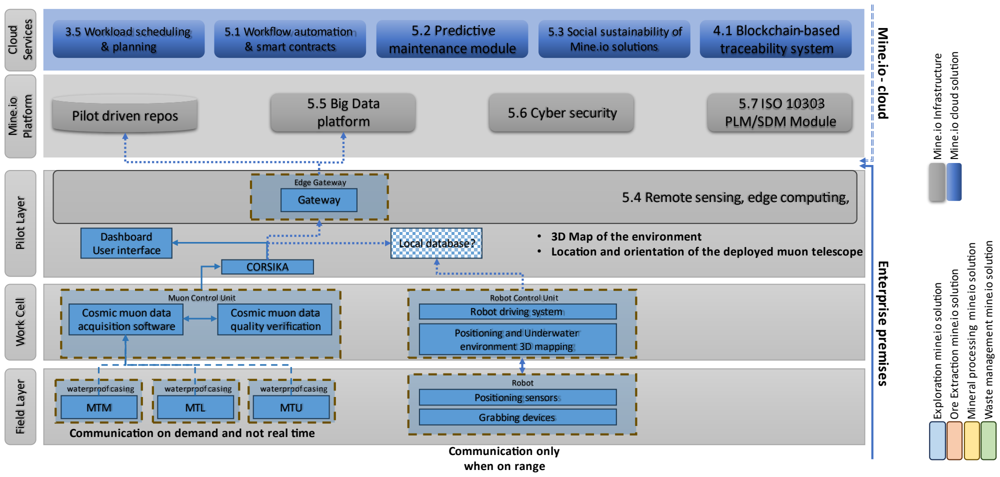

Figure 35: Scenario-Specific Architecture for Use Case 6.1

The current architecture for Use Case 6.1 is illustrated in Figure 35 and relationship is as follows. The objective with the Use Case 6.1 is to record muons at different location in the flooded mine. The muon telescopes detect cosmic muons and collect data on their arrival times and energies, while the waterproof casings protect them from the harsh underwater environment. Positioning sensors determine the location and orientation of the muon telescopes, and underwater vehicle is used to move the muon telescope in the underwater environment. Data from the muon telescope is analysed and thereafter send to the Mine.io big data platform. The position data is transmitted to the Robot Control Unit and local database for further processing. Communication between the Field Layer components and the Mine.io platform layer may be limited by the range of the communication technologies used, and it may not be continuous but rather triggered by specific events or on-demand requests.

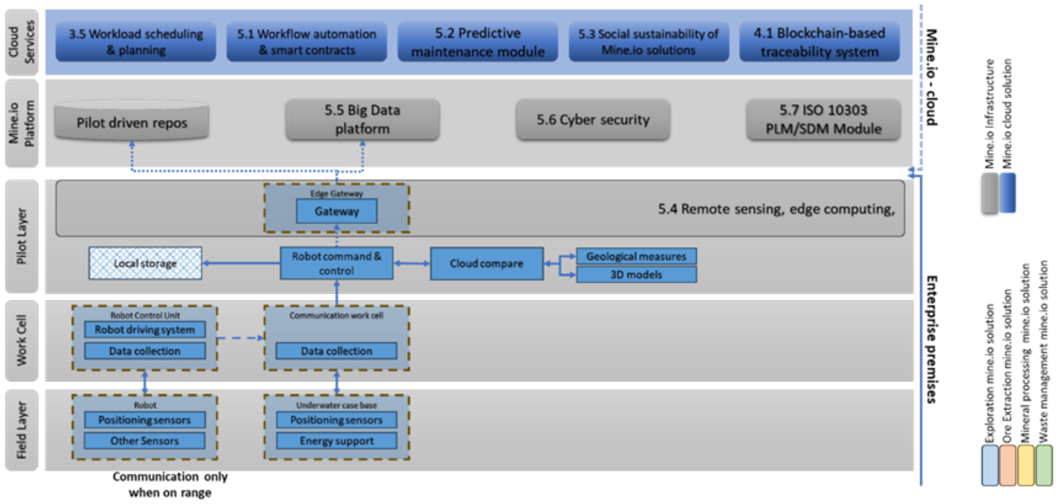

Figure 36: Scenario-Specific Architecture for Use Case 6.2

The current architecture for Use Case 6.2 is illustrated in Figure 36 and relationship is as follows. The objective with the Use Case 6.2 is to inspect and explore the flooded mine. This system contains robot control units, robot driving systems, data collection modules, positioning sensors, energy support systems, and other sensors. The robot control units coordinate the operation of the robots, while the robot driving systems control their movement. Data collection modules collect data from the robots and their sensors, such as positioning data, environmental measurements, and energy usage. Positioning sensors determine the location and orientation of the robots, while energy support systems provide power to the robots. Other sensors may be used to collect additional data, such as camera, temperature and pressure. The collected data is transmitted to the Mine.io Platform Layer for further processing and analysis, enabling the system to monitor and understand the underwater environment. Communication between the Field Layer components and the Platform Layer may be limited by the communication technologies used, and it may not be continuous but rather triggered by specific events or on-demand requests.

7 DEVELOPMENT AND DEPLOYMENT INFRASTRUCTURE

7.1 OVERVIEW

DevOps represents a change in IT culture, focusing on rapid IT service delivery through the adoption of agile, lean practices in the context of a system-oriented approach. DevOps emphasizes people (and culture), and seeks to improve collaboration between operations and development teams.

The practical implementation of DevOps goes through the CI/CD processes, which are delivered through the combined practices of Continuous Integration (CI) and Continuous Delivery (CD).

The Infinitech Way as a service (IWaas) is a GFT service exploiting the results of the H2020 INFINITECH project and providing a complete CI/CD infrastructure and blueprints to support dev/ops activities for innovation projects.

Mine.io Consortium decided to adopt IWaas for the development and deployment of the cloud services of the project.

7.2 ARCHITECTURE

The IWaas infrastructure, which will be used as the DevOps infrastructure of the project, is presented in Figure 37 below.

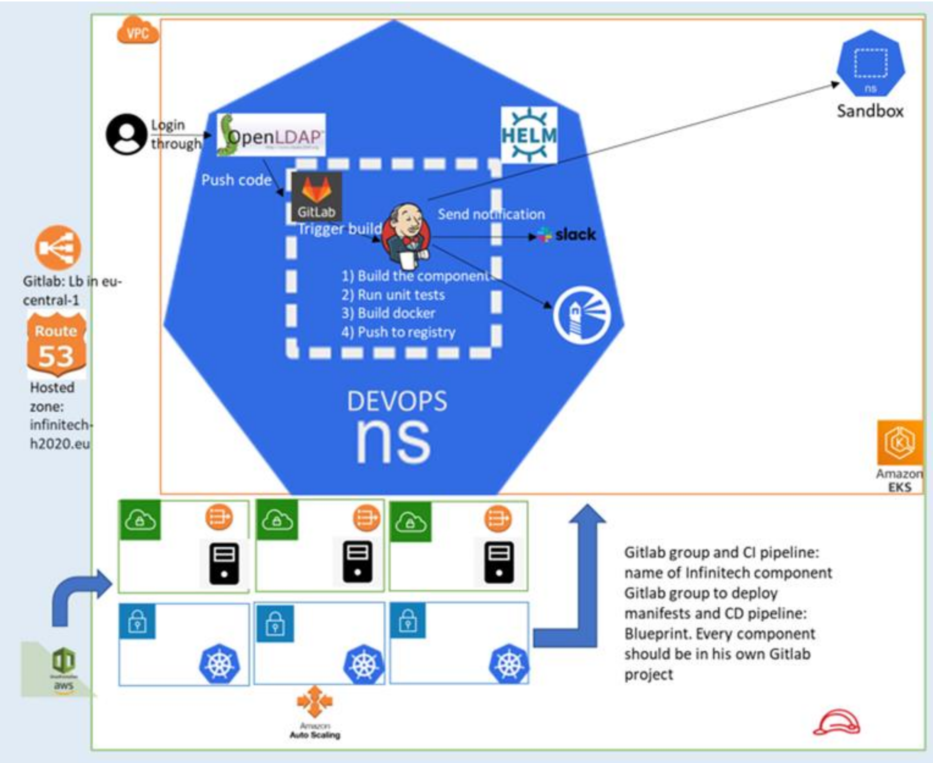

Figure 37: DevOps Infrastructure

The IWaas infrastructure is based on AWS Cloud and consists of the following tools:

- GitLab: a repository application for code version management
- Jenkins: a server for CI/CD pipelines automation
- Harbor: a docker container repository
- Kubernetes (K8s): an automatic deployment and scaler of application containers based on a cluster of nodes
- OpenLDAP: an application to manage organizations and users with LDAP
- Route53: an application to manage domains and DNS records

A specific PROJECT GROUP in GitLab and a specific group in OpenLDAP have been assigned to Mine.io project.

7.3 Holistic and Agile Development

Following a holistic, agile and iterative approach in the design and development processes, all the business requirements coming from the Mine.io contract and the Pilots' use cases (reported in chapter 3) will be collected in the form of user stories for the Mine.io Platform and then analysed to produce the technical requirements driving the development.

REQUIREMENTS MANAGEMENT

Exploiting GitLab features the following process will be followed:

- All the user stories will be recorded as issues in a single "Business Requirements" project (the product backlog) and categorized through the following labels:
  - Origin (Use case #/Mine.io contract)
  - Collecting task (T2.2/T2.3/T2.4)
  - Functional/Non functional
- For each component in the Development view, a project will be created and the technical requirements will be recorded as technical issues, categorized through a label reporting the development task and linked to the user stories
- Once all the user stories and the related technical issues are recorded and categorized with the labels, they will be analysed and divided in different development sprints assigning a milestone.

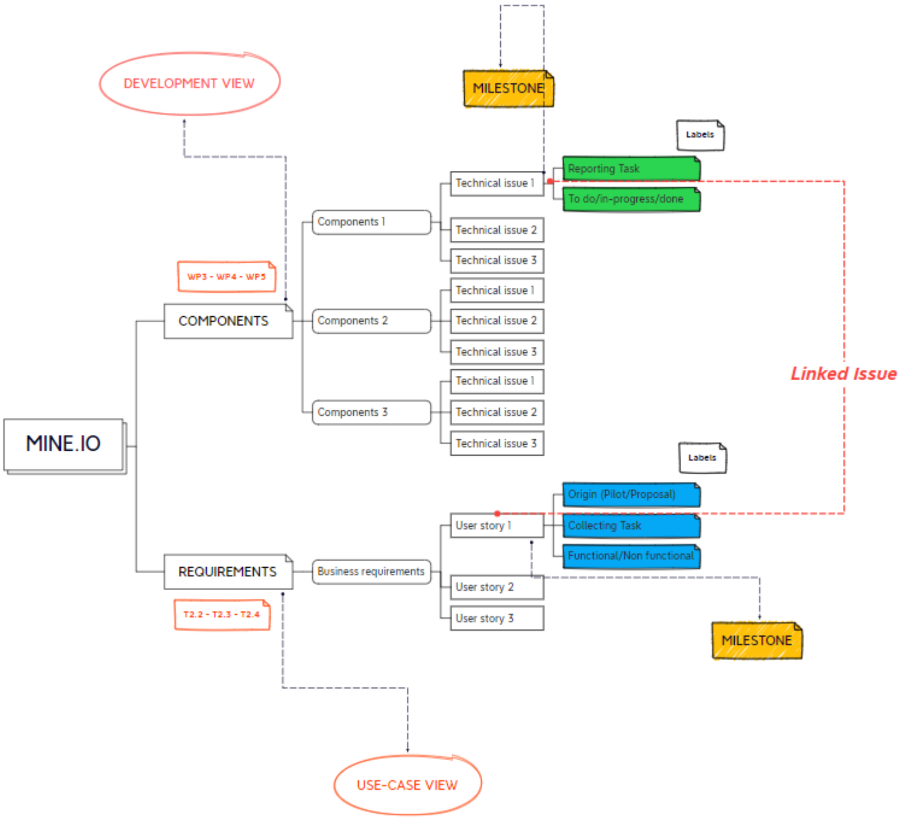

Figure 38: Holistic Requirements Management

DEVELOPMENT MANAGEMENT

The development process starts, selecting the issues related to the first milestone (sprint) and assigning them the "in dev" label. For each component the selected issues will be implemented and then marked with the "in review" label.

GitLab issue boards provide a central point for collaboration and for monitoring the issue lifecycle. They allow the users to have different views: by Use Case, by Component, by Task.

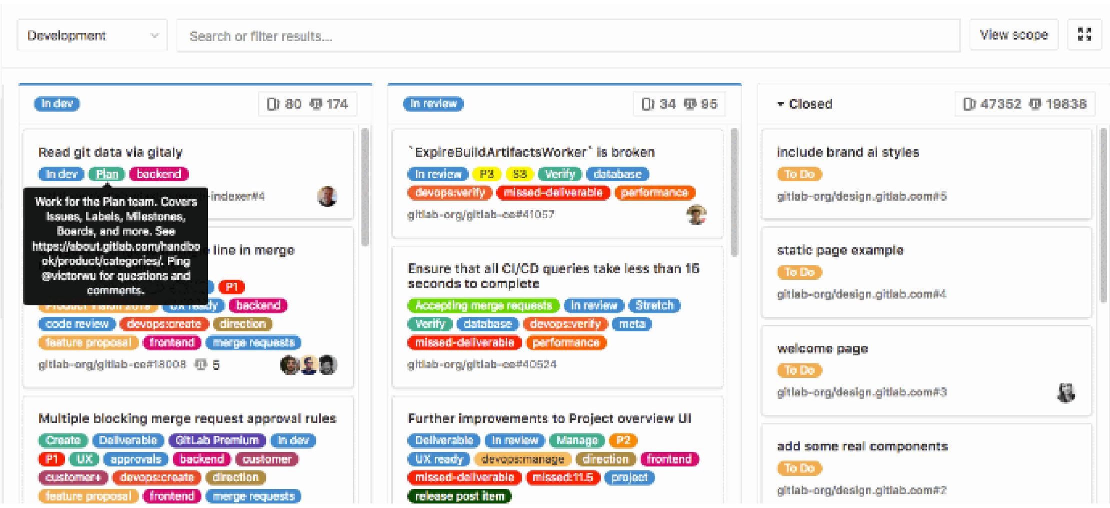

Figure 39: Issue board

7.4 Deployment

Each time a technical issue is completed, and the code updated, a new version of the component is released, integrated with the other components and distributed in the test environment. To enable the CI/CD process the following files must be present in the GitLab repository for each component:

- Dockerfile: defines the instructions for the CI pipeline to build the image
- Jenkinsfile: provides instructions for the CI pipeline
- Jenkinsfile_deploy: provide the instructions for deploy
- The Jenkinsfile enables the user to create a Docker image, push it on the container registry provided by GFT and update the user vie email about the build status.
- The repository provides a Jenkinsfile template. The user can copy the Jenkinsfile template (available in one of the blueprint projects) and apply some minor but very important modifications.

Once these files are created, GitLab can automatically deploy the components in the integrated environment.

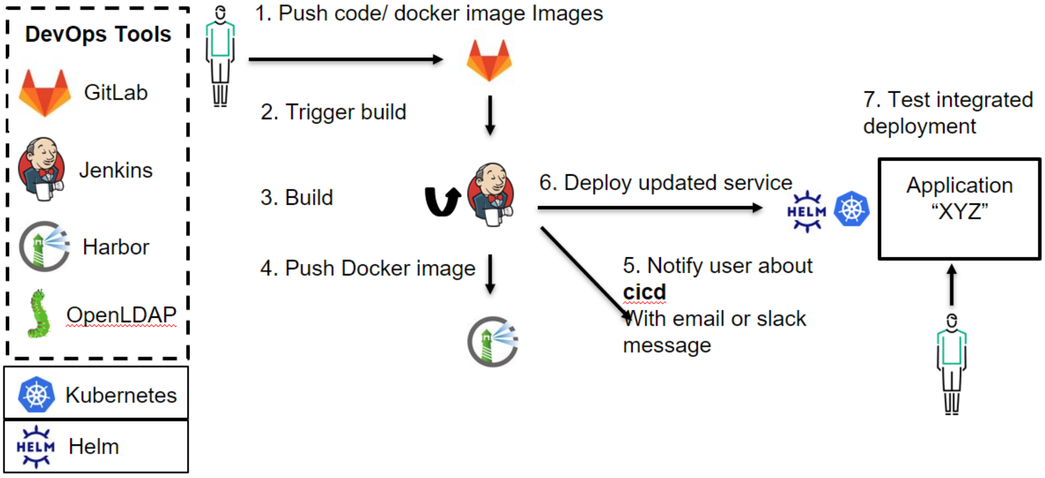

Figure 40: CI/CD Process

When a developer pushes new component code, Gitlab invokes a webhook on Jenkins, which starts any job affected by the code changes. The job builds the component, runs unit tests and, if everything has worked in a proper way, builds an updated Docker image and pushes it to Nexus. In order to deploy as required, the component Helm manager will be used. At the end of the process, Jenkins sends a notification to a dedicated CI/CD channel on the Mine.io Slack project, so that developers are informed that a new build occurred and whether it was successful or not. In case of errors, developers will have to inspect the build logs, find the problem and correct it. In case of success, developers will go ahead and test that the new version works correctly in the test environment. Each Pilot will have its own dedicated Namespace in the Kubernetes cluster providing an isolated test environment (Sandbox). Pilots' users will use the Sandbox to validate the user stories. Once a user story is validated it is marked as "done" as well as all the corresponding technical issues.

8 CONCLUSIONS, OUTLOOK AND NEXT STEPS

The development of an architecture for a digital mining ecosystem is generally challenging considering the complexity of the different mining processes and technologies involved, the need to converge IT/OT, as well as the quite low degree of digitization of several mining sites.

Mine.io has addressed these challenges by building on existing reference architectures and best practices, while considering feedback, requirements and specifications of the Mine.io pilots and components.

The architecture presented in this deliverable provides a sound basis and valuable input for commencing the Mine.io implementation and integration work in a structured way. However, it is ongoing work of the architecture specification and consideration, as the effort will continue and be updated, while more detailed outcomes will be presented in the final version of the deliverable (D2.5).

The design of the Mine.io architecture has considered different industry-standard High-Level Reference Architectures (HLRA) proposed by organisations, institutes or companies. Useful input has been derived from then into the Mine.io architecture regarding its structure, the stage operations it needs to comprise, the required supplementary functions, and layers for communication and components integration.

Additionally, major input has been received from the Mine.io components and solutions and has been added into the Mine.io architecture concerning the building blocks of the system and the features it needs to accommodate.

Overall, the existing Mine.io architecture presents a simple view which can be utilised by the mining industry in regard to the digitization of its processes and can assist mine operators in customizing their sites towards an interconnected system aiming to improve performance and have better situational awareness of the mining operations on the field.

The Mine.io architecture will be further developed by the input to be received by Mine.io partners who are currently developing their components/solutions, and by end users who will participate in the Mine.io Use Case scenarios design and implementation. The final version (version 3) of the deliverable of the Mine.io architecture will, thus, be even more adapted to the specifics of modern mining industry, will address the need for remote monitoring of the mining operations, and provide useful technical proposals contributing to the urge for digitized and interconnected future mining sites.

References

[Aziz20] Aziz, A.; Schelén, O.; Bodin, U. A Study on Industrial IoT for the Mining Industry: Synthesized Architecture and Open Research Directions. IoT 2020, 1, 529-550. https://doi.org/10.3390/iot1020029

[Cisco20] Industrial Automation in Mining Environments, (2023, August 22), Cisco Book.

https://www.cisco.com/c/en/us/td/docs/solutions/Verticals/Industrial_Automation/IA_Verticals/Mining/IA-Mining-DG.html

[Elbazi22] N. Elbazi, M. Mabrouki, A. Chebak and F. Hammouch, "Digital Twin Architecture for Mining Industry: Case Study of a Stacker Machine in an Experimental Open-Pit Mine," 2022 4th Global Power, Energy and Communication Conference (GPECOM), Nevsehir, Turkey, 2022, pp. 232-237, doi: https://ieeexplore.ieee.org/document/9815618.

[Kruchten95] Philippe Kruchten. 1995. The 4+1 View Model of Architecture. IEEE Softw. 12, 6 (November 1995), 42–50. https://doi.org/10.1109/52.469759

[Martin21] D. Martin, N. Kühl and M. Schwenk, "Towards a Reference Architecture for Future Industrial Internet of Things Networks," 2021 IEEE 23rd Conference on Business Informatics (CBI), Bolzano, Italy, 2021, pp. 1-9, doi: https://ieeexplore.ieee.org/document/9610707.

[Pekol19] Pekol, A. Evaluation and Risk Analysis of Open-Pit Mining Operations. Berg Huettenmaenn Monatsh 164, 232–236 (2019). https://doi.org/10.1007/s00501-019-0854-9

[Rockwell19] "Mining Architecture Guide", Rockwell Automation Publication MIN-AT002A-EN-P – November 2019, available: https://literature.rockwellautomation.com/idc/groups/literature/documents/at/min-at002_-en-p.pdf

[Rožanec22] Jože M. Rožanec, Inna Novalija, Patrik Zajec, Klemen Kenda, Hooman Tavakoli Ghinani, Sungho Suh, Entso Veliou, Dimitrios Papamartzivanos, Thanassis Giannetsos, Sofia Anna Menesidou, Ruben Alonso, Nino Cauli, Antonello Meloni, Diego Reforgiato Recupero, Dimosthenis Kyriazis, Georgios Sofianidis, Spyros Theodoropoulos, Blaž Fortuna, Dunja Mladenić & John Soldatos (2022) Human-centric artificial intelligence architecture for industry 5.0 applications, International Journal of Production Research, DOI: https://www.tandfonline.com/doi/full/10.1080/00207543.2022.2138611

[Soldatos21] John Soldatos, Oscar Lazaro, Franco Cavadini (eds.), "The Digital Shopfloor: Industrial Automation in the Industry 4.0 Era", River Publishers Series in Automation, Control and Robotics, ISBN: 9788770220415, e-ISBN: 9788770220408, doi: https://doi.org/10.13052/rp-9788770220408, April 2019

[Young19] Young, A.; Rogers, P. A Review of Digital Transformation in Mining. Min. Metall. Explor. 2019, 36, 683–699.

[ISO] ISO/IEC/IEEE Systems and software engineering -- Architecture description; https://standards.ieee.org/

[OMG] Object Management Group® Standards Development Organization (OMG® SDO) , Data Distribution Service , https://www.omg.org/spec/DDS/

[OPC-UA] OPC Foundation, Open Platform Communications (OPC), Unified Architecture (UA)

https://opcfoundation.org/about/opc-technologies/opc-ua/